



**浙江大學**  
ZHEJIANG UNIVERSITY

**PriceHub 好价汇**

**商品价格比较网站**

**B/S Program Design Report**

姓名 王淳

学号 3220105023

# 目 录

第 1 章 项目描述 .....	1
第 2 章 项目需求 .....	2
2.1 功能性需求 .....	2
2.2 非功能性需求 .....	3
第 3 章 开发环境 .....	4
3.1 操作系统与硬件配置 .....	4
3.2 软件环境 .....	4
3.3 开发过程代码规范限制 .....	4
3.4 技术框架 .....	4
3.4.1 前端技术框架 .....	4
3.4.2 后端技术框架 .....	5
第 4 章 数据库 .....	7
4.1 User .....	7
4.2 Category .....	7
4.3 Good .....	7
4.4 Version .....	7
4.5 Cart .....	8
4.6 ER 图 .....	8
第 5 章 前后端接口 .....	9
第 6 章 使用与测试记录 .....	11
6.1 用户管理 .....	11
6.2 商品管理 .....	11
6.3 信息推送 .....	11
第 7 章 心得与体会 .....	13

图目录

图 2-1 基础对象架构图	2
图 3-1 PriceHub 架构图	5
图 4-1 数据库设计包括五张表: User 表存储用户基本信息, Category 表记录商品类别 (支持多级分类), Good 表存储商品信息, Version 表记录商品的不同版本 (包括价格和创建时间), Cart 表记录用户的购物车内容。Cart 表通过外键关联 User、Good 和 Version 表, 表示用户在购物车中添加的具体商品和版本信息。	8
图 6-1 用户界面组图: 展示了注册、登录和找回密码的界面。展示了用户管理系统的核心功能界面, 包括注册、登录和找回密码。注册界面允许用户填写用户名、密码和邮箱, 系统会验证用户名和密码长度是否大于 6 字节, 邮箱格式是否正确, 并确保用户名和邮箱在系统中唯一。登录界面用于验证用户身份, 用户输入用户名和密码后, 系统会进行匹配, 成功则进入用户主页。找回密码界面帮助用户通过注册邮箱重置密码, 系统会验证邮箱是否存在并发送重置验证码, 确保用户账户的安全性和易用性。设计较为简洁直观, 功能完善, 为用户提供了流畅的操作体验。	11
图 6-2 商品管理界面: 展示了全部历史商品、特定平台筛选、更新商品前后的界面变化。	11
图 6-3 邮件与商品管理界面以及价格变化界面: 展示了收藏夹商品列表、提醒邮件、验证码邮件以及价格变化的功能界面。	12

表目录

表 4-1 User 表	7
表 4-2 Category 表	7
表 4-3 Good 表	7
表 4-4 Version 表	7
表 4-5 Cart 表	8
表 5-1 用户相关接口	9
表 5-2 版本相关接口	9
表 5-3 分类相关接口	9
表 5-4 商品相关接口	10
表 5-5 购物车相关接口	10

## 第 1 章 项目描述

在本学期的《B/S 体系软件设计》课程上，我开发了 PriceHub 好价汇，一个智能比价平台。这个项目旨在帮助用户在主流电商平台中快速找到心仪商品的最低价格，提供便捷、高效的购物体验。

PriceHub 好价汇不仅实现了用户注册与登录、商品价格查询、商品库管理、商品查询与历史价格分析、降价提醒等核心功能，还特别注重移动端适配，确保用户在手机上也能获得友好的使用体验。

通过这个项目，我深入理解并掌握了 Web 应用开发技术和开发的总体流程，包括前端开发、后端开发、数据库设计与管理、Web API 的使用等。我还编写了完整的软件项目文档，记录了项目的开发过程和关键技术点。

## 第 2 章 项目需求

### 2.1 功能性需求

#### • 用户管理：

- **用户注册与登录：**用户可以注册新账号并登录，注册时需验证用户信息，如用户名、密码要求在 6 字节以上，email 的格式验证，并保证用户名和 email 在系统中唯一。
- **用户信息管理：**用户可以修改个人信息，如密码、邮箱等。

#### • 商品价格查询：

- **商品名称查询：**用户登录后，可以通过商品名称查询多个电商平台的实时价格。商品名称建议分词处理优化查询。
- **多平台比价：**支持至少 2 个以上平台查询价格进行比较（淘宝、京东等）。
- **商品库管理：**建立商品库，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，方便后续查询。

#### • 商品查询界面：

- **商品信息显示：**提供商品查询界面能显示商品信息，包括名称、图片、规格等。
- **历史价格图表：**把历史价格用图表形式显示（如价格走势图）。

#### • 购物车内商品降价提醒：

- **降价提醒设置：**支持设置降价提醒，针对指定商品（用户添加到购物车的商品）定时查询最新价格，如有降价发送提醒。

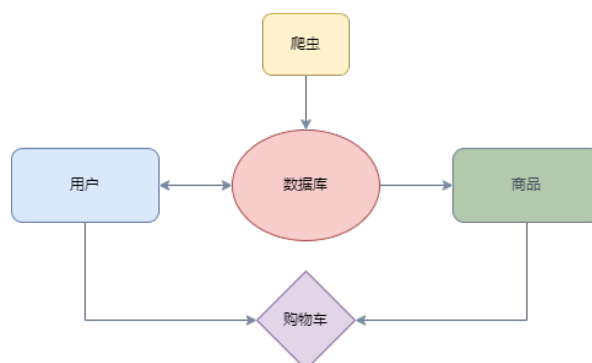


图 2-1 基础对象架构图

- **提醒方式：**可以通过邮件，App 推送等方式实现。

- **移动端适配：**

- **样式适配：**样式适配手机，开发手机 App 或能够在手机浏览器/微信等应用内置的浏览器中友好显示。
- **增强功能：**如开发手机端，可以用相机拍摄商品图片或输入名字进行商品查询。

## 2.2 非功能性需求

- **性能需求：**

- **响应时间：**系统应保证用户查询商品价格的响应时间在合理范围内，通常不超过 3 秒。
- **并发处理：**系统应能处理一定数量的并发用户请求，确保在高并发情况下系统稳定运行。

- **安全性需求：**

- **用户数据保护：**用户注册信息、登录信息等敏感数据需进行加密存储和传输，防止数据泄露。
- **访问控制：**系统应实现用户权限管理，确保只有授权用户才能访问特定功能。

- **可用性需求：**

- **界面友好：**系统界面应简洁、直观，方便用户操作。
- **错误处理：**系统应提供友好的错误提示信息，帮助用户快速解决问题。

- **可维护性需求：**

- **代码规范：**代码应遵循统一的编码规范，便于后续维护和扩展。
- **文档完善：**提供完整的项目文档，包括设计文档、使用手册、测试报告等，方便后续维护和升级。

- **可扩展性需求：**

- **模块化设计：**系统应采用模块化设计，方便后续功能的扩展和升级。
- **平台兼容性：**系统应支持多种主流电商平台，并能够方便地添加新的平台支持。

## 第 3 章 开发环境

### 3.1 操作系统与硬件配置

- **操作系统：** WSL Ubuntu 22.04。
- **硬件配置：** 具备 4 核 CPU、8GB 内存及以上。

### 3.2 软件环境

- **Web 服务器：** 使用 Nginx 1.10 或以上版本作为 Web 服务器。
- **前端框架：** 使用 npm 10.8.1 及以上的版本以及 Vue3, Element UI
- **后端框架：** 使用 Spring Boot 2.6 或以上版本，以及对应版本的 Spring Data JPA 等库。
- **数据库：** 使用 MySQL 8.0 或以上版本，并配置好相应的数据库连接。
- **网络环境：** 使用 HTTP 协议对系统进行传输。

### 3.3 开发过程代码规范限制

- **代码风格：** 遵循 Google Java 编程规范，使用 IntelliJ IDEA 或 VScode 等 IDE 进行开发。
- **避免使用过时、不安全的技術：** 尽量使用最新的技术，避免使用过时的技术，如使用过时的 Spring 版本等。
- **代码注释：** 代码中需要添加详细的注释，方便其他开发者阅读和理解。
- **严格控制代码质量：** 避免出现内存泄漏、缓冲区溢出、SQL 注入等安全漏洞。
- **采用版本控制工具：** 如 Git 等，方便版本控制。

### 3.4 技术框架

#### 3.4.1 前端技术框架

本项目前端采用 Vue + Element UI + npm 包管理工具作为主体框架：

- **Vue：** Vue.js 是一款用于构建用户界面的渐进式 JavaScript 框架。它以其简洁的 API、灵活的组合方式和高效的渲染性能而受到广泛关注。Vue 具有响应式数据

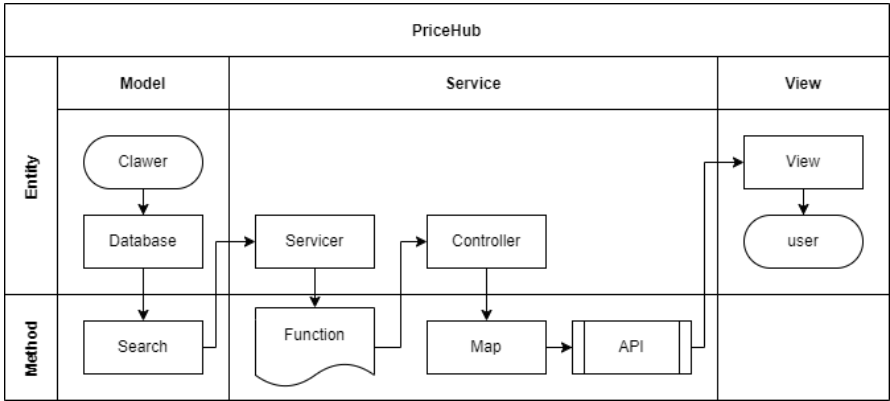


图 3-1 PriceHub 架构图

绑定、组件化开发、虚拟 DOM 等特性，使得开发者能够快速构建出交互性强、性能优良的单页应用。Vue 的渐进式设计理念允许开发者根据项目需求逐步引入和扩展功能，无论是小型项目还是大型复杂应用都能轻松应对。

- **Element UI：**Element UI 是一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的桌面端组件库。它提供了丰富的 UI 组件，如按钮、表单、表格、弹窗等，帮助开发者快速搭建美观、易用的界面。Element UI 遵循一致性、实用性、通用性等设计原则，提供了简洁的 API 和详细的文档，使得开发者能够轻松上手并高效开发。
- **npm 包管理工具：**npm（Node Package Manager）是 Node.js 的默认包管理工具，用于管理 JavaScript 代码库和项目依赖。它为开发者提供了以下功能：包管理，npm 允许开发者轻松地安装、更新和删除项目的依赖包。它支持一个庞大的开源包库，开发者可以通过 npm 下载和使用现成的工具和库；依赖管理，npm 通过 package.json 文件记录项目的依赖，确保所有开发者和部署环境都能使用相同的版本。开发者可以使用 npm install 来自动安装这些依赖。

3.4.2 后端技术框架

本项目后端采用 Java + Spring Boot + Springboot MVC + Spring Data JPA 作为主体框架，同时采用 MySQL 作为数据库：

- **Java：**Java 是一种广泛使用的编程语言，以其跨平台性、面向对象设计、安全性和可靠性而闻名。在企业级应用开发中，Java 占据重要地位，拥有丰富的类库和强大的生态系统。其简洁明了的语法使得学习和使用变得容易，同时其卓越的性能和稳定性使其成为后端开发的首选语言之一。
- **Spring Boot：**Spring Boot 是由 Pivotal 团队开发的一个新型框架，旨在简化新 Spring 应用的初始设置和开发流程。通过采用特定的配置方式，Spring Boot 消除



了开发者编写样板化 XML 文件的需求。凭借其简化配置、快速开发和灵活扩展的特性，Spring Boot 已成为 Java 后端开发的主流选择，广泛应用于微服务、企业级应用和 Web 开发领域。

- **Spring MVC：** Spring MVC 是 Spring 框架的一部分，专门用于构建 Web 应用程序。它遵循模型-视图-控制器（MVC）设计模式，将业务逻辑、数据展示和用户交互分离，从而提高代码的可维护性和可扩展性。Spring MVC 提供了丰富的功能，如请求映射、数据绑定、表单验证和视图解析，使开发者能够快速构建功能强大的 Web 应用。
- **Spring Data JPA：** Spring Data JPA 是 Spring Data 项目的一个子模块，旨在简化与 JPA（Java Persistence API）数据访问层的集成。它通过提供高级抽象，减少了重复代码，使开发者能够更便捷地进行数据库操作。Spring Data JPA 显著减少了与数据库交互时的模板代码，使开发者能够专注于业务逻辑的实现，从而提高开发效率。它特别适合快速开发数据驱动的应用。
- **MySQL：** MySQL 是一个开源的关系型数据库管理系统（RDBMS），广泛应用于 Web 应用和数据分析领域。凭借其高性能、可靠性和易用性，MySQL 成为全球最流行的数据库之一。它的高效、可靠和易用特性使其成为从小型到中大型应用的数据库首选，特别适合 Web 开发和数据驱动的应用程序。
- **爬虫：** 后端还集成了由 Python 或者 Java 实现的爬虫功能。目前正在同步测试两种实现方法，具体安排在后续安排中，将会使用爬虫不断更新数据库并且获得及时的最新的数据。

## 第 4 章 数据库

### 4.1 User

编号	属性名	字段名称	数据类型	长度	备注
1	用户 ID	id	VARCHAR	255	主键
2	用户名	name	VARCHAR	255	不空
3	密码	password	VARCHAR	255	不空
4	邮箱	email	VARCHAR	255	不空

表 4-1 User 表

### 4.2 Category

编号	属性名	字段名称	数据类型	长度	备注
1	类别 ID	id	BIGINT	20	主键, 自增
2	类别名称	name	VARCHAR	255	不空
3	父类别 ID	parent_id	BIGINT	20	外键

表 4-2 Category 表

### 4.3 Good

编号	属性名	字段名称	数据类型	长度	备注
1	商品 ID	id	BIGINT	20	主键, 自增
2	商品名称	name	VARCHAR	255	不空
3	商品描述	description	VARCHAR	255	
4	商品类别	category	VARCHAR	255	

表 4-3 Good 表

### 4.4 Version

编号	属性名	字段名称	数据类型	长度	备注
1	版本 ID	id	BIGINT	20	主键, 自增
2	商品 ID	goodId	BIGINT	20	外键
3	创建时间	createdAt	DATETIME		
4	价格	price	DOUBLE		

表 4-4 Version 表

4.5 Cart

编号	属性名	字段名称	数据类型	长度	备注
1	版本 ID	versionId	BIGINT	20	外键
2	商品 ID	goodId	BIGINT	20	外键
3	用户 ID	userId	VARCHAR	255	外键

表 4-5 Cart 表

4.6 ER 图

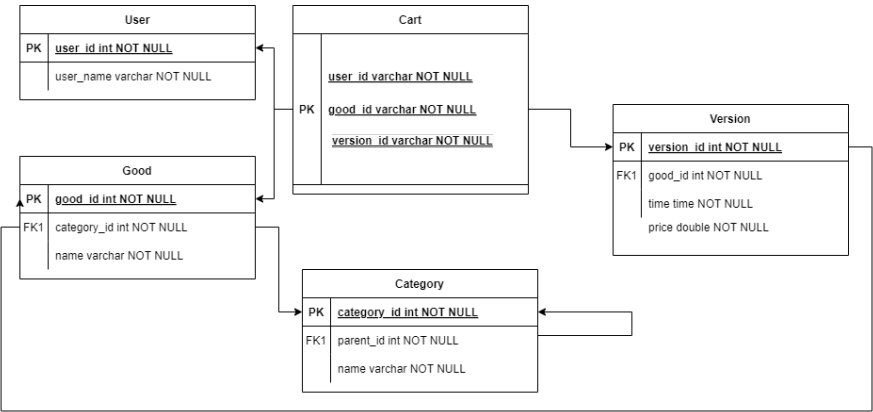


图 4-1 数据库设计包括五张表：User 表存储用户基本信息，Category 表记录商品类别（支持多级分类），Good 表存储商品信息，Version 表记录商品的不同版本（包括价格和创建时间），Cart 表记录用户的购物车内容。Cart 表通过外键关联 User、Good 和 Version 表，表示用户在购物车中添加的具体商品和版本信息。

第 5 章 前后端接口

功能	URL	请 求 方式	请求参数	返回信息
查 看 详情	/api/user/create	POST	User 对象	成功: msg: "success", code: 200, data: 用户数据
查 看 列表	/api/user/all	GET	无	成功: msg: "success", code: 200, data: 用户列表
登录	/api/user/login	POST	LoginRequest 对象	成功: msg: "login successful", code: 200, data: "登录成功" 失败: msg: "invalid credentials", code: 401, data: "凭证无效"
查 询 用户	/api/user	POST	QueryRequest 对象	成功: msg: "success", code: 200, data: 用户数据 失败: msg: "user not found", code: 404, data: null

表 5-1 用户相关接口

功能	URL	请 求 方式	请求参数	返回信息
创 建 版本	/api/versions/create	POST	Version 对象	成功: msg: "success", code: 201, data: 版本数据
查 询 版本	/api/versions/query/goodId	GET	goodId 商品 ID	成功: msg: "success", code: 200, data: 版本列表

表 5-2 版本相关接口

功能	URL	请 求 方式	请求参数	返回信息
创 建 分类	/categories/create	POST	name (分类名称), parentId (可选, 父分类 ID)	成功: msg: "Category created successfully!"
获 取 所 有 分类	/categories	GET	无	成功: msg: "success", code: 200, data: 分类列表

表 5-3 分类相关接口

功能	URL	请求方式	请求参数	返回信息
创建商品	/api/goods	POST	Good 对象	成功: msg: "success", code: 201, data: 商品数据
查看所有商品	/api/goods/all	GET	无	成功: msg: "success", code: 200, data: 商品列表
获取商品信息	/api/goods/id	GET	无	成功: msg: "success", code: 200, data: 商品数据
更新商品	/api/goods/id	PUT	Good 对象	成功: msg: "success", code: 200, data: 更新后的商品数据
删除商品	/api/goods/id	DELETE	无	成功: msg: "success", code: 204, data: null

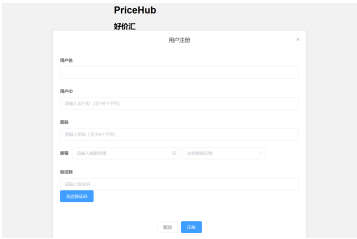
表 5-4 商品相关接口

功能	URL	请求方式	请求参数	返回信息
添加商品到购物车	/api/cart/add	POST	userId(用户 ID), goodId(商品 ID), quantity(数量)	成功: msg: "商品添加成功", code: 200
获取购物车内容	/api/cart/id	GET	无	成功: msg: "成功", code: 200, data: 购物车商品列表
更新购物车商品数量	/api/cart/update	PUT	userId(用户 ID), goodId(商品 ID), quantity(新数量)	成功: msg: "商品数量更新成功", code: 200
删除购物车商品	/api/cart/remove	DELETE	userId(用户 ID), goodId(商品 ID)	成功: msg: "商品删除成功", code: 200
清空购物车	/api/cart/clear	DELETE	userId(用户 ID)	成功: msg: "购物车已清空", code: 200

表 5-5 购物车相关接口

# 第 6 章 使用与测试记录

## 6.1 用户管理



(a) 注册界面



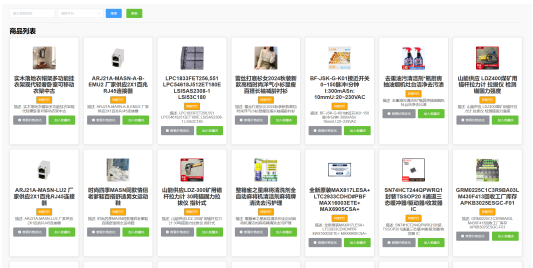
(b) 登陆界面



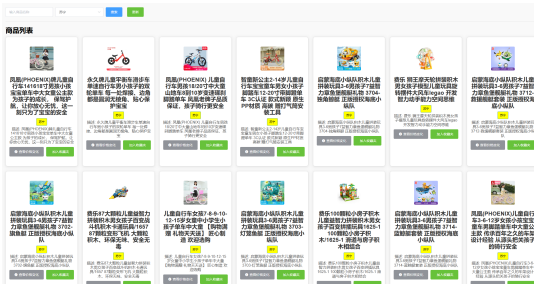
(c) 找回密码

图 6-1 用户界面组图：展示了注册、登录和找回密码的界面。展示了用户管理系统的核心功能界面，包括注册、登录和找回密码。注册界面允许用户填写用户名、密码和邮箱，系统会验证用户名和密码长度是否大于 6 字节，邮箱格式是否正确，并确保用户名和邮箱在系统中唯一。登录界面用于验证用户身份，用户输入用户名和密码后，系统会进行匹配，成功则进入用户主页。找回密码界面帮助用户通过注册邮箱重置密码，系统会验证邮箱是否存在并发送重置验证码，确保用户账户的安全性和易用性。设计较为简洁直观，功能完善，为用户提供了流畅的操作体验。

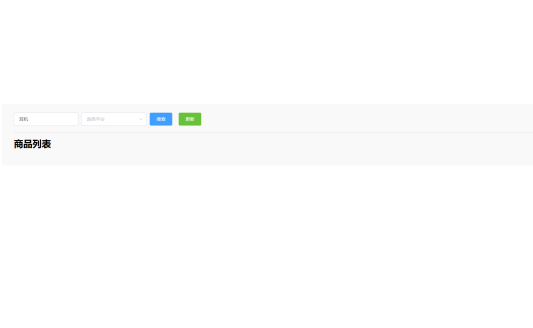
## 6.2 商品管理



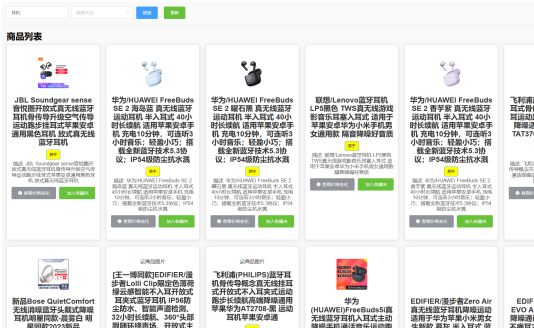
(a) 全部历史商品界面



(b) 特定平台筛选



(c) 更新商品前



(d) 更新商品后

图 6-2 商品管理界面：展示了全部历史商品、特定平台筛选、更新商品前后的界面变化。

## 6.3 信息推送



## 第 7 章 心得与体会

完成这个商品比价网站的 BS 程序，对我来说是一次充满挑战和收获的经历。作为单人开发的项目，我从需求分析、技术选型到编码实现、测试部署，全程独立完成，这让我对全栈开发有了更深刻的理解。尽管项目达到了基本的功能目标，但也有一些遗憾，比如由于时间有限，未能实现管理员功能和个性化推荐功能。

在开发过程中，我深刻体会到了时间管理和优先级划分的重要性。为了实现核心的比价功能，我不得不将一些附加功能（如管理员后台和个性化推荐）暂时搁置。虽然这是一个无奈的选择，但也让我学会了如何在有限的时间内聚焦于最关键的任务，确保项目能够按时交付并具备可用性。

值得一提的是，这个项目的核心功能——商品比价，离不开爬虫技术的支持。在这里，我要特别感谢胡老师。在课堂上，胡老师详细演示了爬虫的实现原理和技巧，这为我开发比价功能提供了极大的帮助。通过学习老师的示例代码和思路，我了解了实现爬虫的基本技术。

为了进一步提升网站的性能和用户体验，我在项目中应用了异步编程和多线程技术。在爬取商品数据时，传统的同步方式会导致程序阻塞，效率低下。通过使用异步编程，我能够同时发起多个网络请求，显著提高了数据抓取的效率。此外，在处理大量数据时，我引入了多线程技术，将任务分配到多个线程中并行执行，进一步优化了程序的响应速度和处理能力。这些技术的应用不仅让网站的性能得到了显著提升，也让我对高并发编程有了更深的理解。

总的来说，这次项目让我在技术能力、时间管理和问题解决等方面得到了全面的提升。虽然有一些功能未能实现，但这也让我更加清楚地认识到自己的不足和改进空间。比如后端代码，我并没完全按照最初的 MVC 思路编写，而是按类来编写，而且有时代码简洁性仍然有待提升，可以更加优雅。我相信，随着不断的学习和实践，我能够逐步完善这个项目，让它变得更加成熟和强大。这段经历不仅是一次技术的锻炼，更是一次成长的旅程，让我对未来充满了信心和动力。

最后，再次感谢胡老师的悉心指导和无私分享，以及对技术栈选择的包容度，作为一个图灵班的学生，其实我们的培养方案并没有要求我们要修开发相关的课程，但是我个人认为无论 AI 技术如何飞跃发展，这些实实在在的开发都极其具有学习价值，也是社会的刚需。