**江苏联合职业技术学院**

**无锡机电分院**

**毕业设计**

毕业论文

|  |  |
| --- | --- |
| 课题名称： | **基于JavaWeb健康咨询网站的设计与实现** |
| 系部： | **电子信息工程系** |
| 专业： | **软件技术** |
| 班级： | **2002** |
| 姓名： | **王俊杰** |
| 课题主持人： | **王俊杰** |
| 组成员： | **高健博 解明辉 许云泓** |
| 指导教师： | **周邵锦** |

2024年11月27日

# 摘 要

经历了全球三年新型冠状病毒爆发所给全人类带来的安全及生命灾害之后，后疫情时代的公众对于自我身体安全和健康养生知识投入了前所未有的精力、时间、财力等等。其中的目的十分明确并且一致：提高抵抗力、增强个人体质、学习健康知识……同时，随着社会和时代的快速前进，各行各业的电子化、信息化程度皆与日俱增。其中，医疗及医学行业的信息集成化和服务高效率化，也成为了国内社会的必然趋势。基于此点，本论文提出了一种基于JavaWeb技术为主的名为“医知便民”的健康咨询网站。该技术广泛应用于各类网页的开发中，具有可靠、使用等特点。

该论文首先阐述了项目的研究背景、项目研究意义、项目研究目的以及研究内容；随后，本文就项目开发的关键技术进行了分析，即相关主流的网络前端开发技术；紧接着，论文就用户的潜在需求进行了分析，确保了项目的可行性；除此之外，本文还涉及了系统概要设计、详细设计与实现以及测试与维护。通过本项目，用户可以获得更加智能、安全、合理的健康知识，为生命安全添砖加瓦。

**关键词：**健康咨询网站；JavaWeb；OpenAI

**Abstract**

After three years of the global outbreak of a new coronavirus, which brought about a safety and life disaster to all mankind, the public in the post epidemic era has invested an unprecedented amount of energy, time, and financial resources in their own physical safety and health and wellness knowledge. The purpose is clear and consistent: to improve resistance, enhance personal fitness, and learn health knowledge. ...... At the same time, with the rapid advancement of society and the times, the degree of electronic and information technology in all walks of life is increasing day by day. Among them, the integration of information and the high efficiency of services in the medical and medical industry have also become an inevitable trend in the domestic society. Based on this, this thesis proposes a JavaWeb technology-based health consultation website called ‘Medical Knowledge for the People’. This technology is widely used in the development of various types of web pages, and it is reliable and easy to use.

The thesis firstly describes the background of the project, the significance of the project, the purpose of the project, and the content of the research; then, the thesis analyses the key technologies of the project development, i.e. the relevant mainstream web front-end development technologies; then, the thesis analyses the potential needs of the users to ensure the feasibility of the project; in addition to that, this thesis also involves the system outline design, the detailed design and implementation, and the testing and maintenance of the system. testing and maintenance. Through this project, users can obtain smarter, safer, and more reasonable health knowledge, which will add to the safety of life.

**Keywords**: health advice website; JavaWeb; OpenAI

**目 录**

[摘 要 I](#_Toc183701679)

[Abstract II](#_Toc183701680)

[第一章 绪论 1](#_Toc183701681)

[1.1 研究背景 1](#_Toc183701682)

[1.2 研究意义 1](#_Toc183701683)

[1.3 研究目的 2](#_Toc183701684)

[1.4 研究内容 3](#_Toc183701685)

[1.5 组内分工 4](#_Toc183701686)

[第二章 关键技术 5](#_Toc183701687)

[2.1 HTML&CSS 5](#_Toc183701688)

[2.2 JavaScript 5](#_Toc183701689)

[2.3 JSP 5](#_Toc183701690)

[2.4 B/S架构(浏览器/服务器架构) 5](#_Toc183701691)

[2.5 MVC（Model–View–Controller）模式 6](#_Toc183701692)

[2.6 Maven 6](#_Toc183701693)

[2.7 Servlet 6](#_Toc183701694)

[2.8 JDBC（Java Database Connectivity） 6](#_Toc183701695)

[第三章 需求分析 8](#_Toc183701696)

[3.1 技术可行性 8](#_Toc183701697)

[3.2 经济可行性 8](#_Toc183701698)

[3.3 社会可行性 9](#_Toc183701699)

[第四章 系统概要设计 10](#_Toc183701700)

[4.1 首页 10](#_Toc183701701)

[4.1.1标题头设计 10](#_Toc183701702)

[4.1.2定位功能设计 10](#_Toc183701703)

[4.1.3“新帖速递”设计 13](#_Toc183701704)

[4.1.4导航块功能设计 13](#_Toc183701705)

[4.1.5页脚设计 13](#_Toc183701706)

[4.2 “对症下药”-药物信息一览表 14](#_Toc183701707)

[4.3 “医院评分”界面 14](#_Toc183701708)

[4.4 “病情论坛”界面 15](#_Toc183701709)

[4.5 登录界面 15](#_Toc183701710)

[4.6 注册界面 17](#_Toc183701711)

[4.7 “密码重置”界面 19](#_Toc183701712)

[4.8 AI咨询界面 20](#_Toc183701713)

[4.9 数据库结构 21](#_Toc183701714)

[第五章 系统详细设计与实现 23](#_Toc183701715)

[5.1 首页界面 23](#_Toc183701716)

[5.2 “对症下药”界面 34](#_Toc183701717)

[5.3 医院评分 36](#_Toc183701718)

[5.4 病情论坛 43](#_Toc183701719)

[5.5 AI自查 48](#_Toc183701720)

[5.6 登录界面 53](#_Toc183701721)

[5.7 注册界面 61](#_Toc183701722)

[5.8密码重置界面 64](#_Toc183701723)

[第六章 系统测试与维护 67](#_Toc183701724)

[6.1 测试目的 67](#_Toc183701725)

[6.2 测试结果 67](#_Toc183701726)

[6.2.1前端 67](#_Toc183701727)

[6.2.2 后端 67](#_Toc183701728)

[第七章 总结与展望 69](#_Toc183701729)

[参考文献 70](#_Toc183701730)

[致谢 71](#_Toc183701731)

# 第一章 绪论

## 1.1 研究背景

当下社会，新冠疫情所带来的影响虽已经离去，但留下的阴影仍然存在——疫情期间，相关在线医疗服务的欠发达性，给公众群体带来了极大的不便利性：在线问诊功能不完善致使用户难以对真实身体状况进行判断、信息源头可靠性加剧了谣言的滋生、相关群体对已有的复杂的信息化服务陌生不熟悉，使得日常生活举步维艰……因此，即使疫情问题已不再，一个简洁明了的、科学合理的、具有创新性的医疗网站仍然对于公众和社会而言必不可少。

传统的医疗健康网站，功能性较为单一、创新性不足，并且大量依赖手动操作，效率低下、错误率高。因此，该类现有的设计对于实现用户所有的需求，仍具有明显的距离。而随着信息技术的发展和设计开发的多元化，基于Web端的医疗自查网站的设计性与日俱增，可实现功能也成倍增长。这样的进步，让用户在利用该网站进行日常获取健康信息或身体不适时进行网上自测时更加便利。同时，该设计也高效化、电子化了医疗行业，为日常服务的提供及潜在危机的应对做出可靠贡献。

HTML作为一种常见的创建网页的标准标记语言，常与CSS、JavaScript一起被众多网站用于设计网页、网页应用程序以及移动应用程序的用户界面。具有兼容性强、搜索引擎优化友好、加载速度快等无可替代的优点。因此，本组决定利用HTML&CSS，进行该医疗网站项目的开发、部署、扩展，为安全科学的健康服务提供强大、简洁的技术支持。

## 1.2 研究意义

21世纪，计算机、信息技术发展呈现出井喷式发展的趋势。同时，电子化、信息化也开始大量集成于各行各业，并表现出智能化态势。在2018年，美国人工智能研究室OpenAI发布了其第一代GPT语言模型。2年后，GPT3模型的发布引发了全球关注，许多知名企业和开发者开始将GPT3集成到自己的应用中。本组研究基于JavaWeb的人工智能医疗网站因此也具有更加现实的意义，其意义主要体现在一下几个方面。

首先，简化了患者健康自查的难度：通过广泛且长期使用的HTML&CSS进行网页编程，能够降低查询资料及个人自查的门槛，充分照顾考虑到了社会特殊群体的需求。从而让全体公众的身体健康得到保障。

其次，提高了医疗服务效率：项目网站通过信息集成化、服务智能化的设计，即：设计各个种类药物特性属性网页、药物搜索功能、医院评分功能、论坛信息页面和人工智能自查功能等等，极大的节省了用户的个人时间。用户无需在传统的搜索引擎中进行长时间的搜索并对信息正确性进行判断。若仅仅有轻微症状时，患者也不必前往医院问诊，而是直接利用网站搜索知识并用药，在相应程度上减轻了医疗部门的压力

再次，普及了科学的健康知识：该网页项目中所设计的“病情论坛“模块可以定期更新医疗信息、健康知识、科普文章等等信息，为广大群众提供科学合理、丰富多样的健康内容。该设计对于疾病防治有重要意义。

最后，规范了信息安全：基于JavaWeb的健康自查网站规范了信息来源的安全性，拓展了功能性。可以刺激同类设计进行相关改进，从而加强医疗信息化、提高服务含金量。

## 1.3 研究目的

该项目研究的主要目的是开发一款基于JavaWeb的健康自查网站。该网站皆在提供便捷、智能、安全、科学的自查服务和AI咨询服务。具体目的包括一下方面：

保障用户的信息安全：通过搭建简单易用、信息准确的网页平台，使用户在日常生活中能更加简单方便地获取健康知识，获取决策支持，扫清决策障碍。

减轻医疗服务的压力：通过项目所设计的“对症下药“板块，提供科学、安全、详细的药品资料，让用户可以根据自身症状和个人判断进行用药。无需前往就医。节约用户时间。，减轻医疗部门压力

传播健康知识：通过项目设计中的“病情论坛“，用户可阅读浏览最新医学知识，对个人信息误区进行纠正、走出信息茧房。

为实现上述目标，为实现上述目标，本研究从理论学习到实践开发展开全面探索，具体内容如下：

学习相关技术与理论：学习并研究开发web项目的相关技术和理论，结合国内外优秀项目的开发过程，有效地分析和设计出一套较完善的健康咨询系统。

功能需求与系统设计：结合对健康咨询系统的功能需求分析，对系统进行适度、合理的 设计，力求实现一套模块化、易开发的业务逻辑层。

动态数据传递与应用表现层开发：根据应用表现层的需求，在业务逻辑层中实现数据的获取、打包和传递，确保资源管理层与应用表现层之间通过中间层实现高效的动态数据交互。

理论与实践结合：通过项目的设计与实现，全面提升软件系统设计与开发能力，将理论知识与实践技能相融合，最终撰写一篇符合软件工程专业要求的毕业论文。

## 1.4 研究内容

1.项目背景与意义分析

鉴于健康需求的日益增长，本项目的设计不仅具有现实性和实践性，同时也具备重要的社会意义。通过研究项目背景，明确了以健康医疗知识自主查询及交流为核心的网站系统的必要性和价值，为系统的设计和实现提供了理论依据。项目将致力于构建一个高效、便捷且符合用户需求的平台。

2.用户需求分析

通过对现有网络设计的分析与总结，项目对优秀模块的优点进行学习，同时整理其问题和缺陷以避免在个人设计中出现。结合用户调研，明确了用户在健康自查、健康信息获取以及AI咨询等方面的需求。具体功能包括“对症下药”的药品信息一览、简洁直观的用户界面、以及系统的可靠性与合理性要求。以上分析为前端设计提供了标准与依据，同时也奠定了设计风格的基础，确保最终的系统能够充分满足用户的实际需求。

3.技术分析与选型

根据项目需求，对相关技术进行了全面的分析和选型。前端开发采用HTML、CSS和JavaScript，旨在快速构建直观、简洁的用户界面。后端开发则基于JavaWeb技术，结合MVC框架构建出快速、稳定且低耦合的B/S三层架构。此技术选型不仅支持功能实现，还在代码维护、扩展性和易用性方面提供了保障。

4.系统设计与实现

在明确需求和技术选型的基础上，制定了系统的具体实现方案。前端开发规划了清晰的功能模块与结构布局，确保用户能够快速找到所需信息；后端则通过稳定的业务逻辑层设计，保障系统运行流畅、数据传输准确。同时，明确了代码的易增、易删、易检查、易更新的设计目标，为后期维护和优化提供支持。

5.系统测试与修改

通过详细的测试方案，对系统功能、界面美观性及结构稳定性进行了全面验证。测试过程中，重点关注设计的正确性和用户体验优化，同时评估系统的扩展性和后期维护能力。通过测试反馈对系统进行改进，确保最终系统实现用户便捷使用和高效运行的目标。

## 1.5 组内分工

在项目开发过程中，本人主要负责前端相关界面的设计和开发，同时协助其他负责前端开发的同学完成界面的美化工作。王俊杰则专注于编写 Java 代码，并负责构建和维护 MySQL 数据库。解明辉和许云鸿共同承担了前端网页的编写任务。全体成员皆为实现项目的功能与效果做出了重要贡献。

# 第二章 关键技术

本项目基于 JavaWeb 技术开发了一套健康自查网站，包含以下关键技术：即HTML&CSS，JavaScript，JSP 技术，B/S架构(浏览器/服务器架构)，MVC，Maven，Servlet，JDBC。

## 2.1 HTML&CSS

HTML 是一种用于定义网页结构的标记语言，通过各种标签来组织内容，如标题、段落、图片和链接等。它是网页的骨架，提供基础的内容展示功能。CSS 是一种样式语言，用于美化和布局 HTML 内容。它通过定义字体、颜色、间距、布局等样式，使网页具有更好的视觉效果和用户体验。CSS 也支持响应式设计，可以根据不同设备调整页面布局。

## 2.2 JavaScript

一种动态的编程语言，广泛应用于网页开发中，主要用于增强网页的交互性和动态性。作为Web开发的核心技术之一，它与HTML和CSS共同协作，HTML负责定义网页的内容结构，CSS处理样式和布局，而JavaScript则控制网页的行为和交互。用户可以利用JavaScript进行事件处理、表单验证、动画创建等多种功能。

## 2.3 JSP

一种动态网页技术标准，全称Java Server Pages。JSP部署于网络服务器上，可以响应客户端发送的请求，并根据请求内容动态地生成HTML、XML或其他格式文档的Web网页，然后返回给请求者。JSP技术以Java语言作为脚本语言，为用户的HTTP请求提供服务，并能与服务器上的其它Java程序共同处理复杂的业务需求。

## 2.4 B/S架构(浏览器/服务器架构)

一种网络架构模式，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。客户端只需要安装一个浏览器，通过Web服务器与数据库服务器进行数据交互。B/S架构利用了Web浏览器技术和Internet协议，实现了异构系统的连接和信息的共享。

## 2.5 MVC（Model–View–Controller）模式

软件工程中的一种软件架构模式，主要作用在项目的业务逻辑层。它把软件系统分为三个基本部分：模型（Model）、视图（View）和控制器（Controller）。MVC模式的目的是实现一种动态的程序设计，简化后续对程序的修改和扩展，并且使程序某一部分的重复利用成为可能。除此之外，MVC模式通过对复杂度的简化，使程序的结构更加直观。软件系统在分离了自身的基本部分的同时，也赋予了各个基本部分应有的功能。专业人员可以通过自身的专长进行相关的分组

## 2.6 Maven

一个基于Java的构建工具，可以帮助开发者快速构建、管理和发布Java项目。它利用了项目对象模型（Project Object Model，POM）来描述项目构建的过程，包括编译、测试、打包等各个阶段，并提供了丰富的插件和依赖机制，实现了对Java项目的自动化管理。

## 2.7 Servlet

运行在 Web 服务器或应用服务器上的程序，它是作为来自 Web 浏览器或其他 HTTP 客户端的请求和 HTTP 服务器上的数据库或应用程序之间的中间层。使用 Servlet，您可以收集来自网页表单的用户输入，呈现来自数据库或者其他源的记录，还可以动态创建网页。

## 2.8 JDBC（Java Database Connectivity）

一种Java API，它允许Java程序与数据库进行交互。JDBC为多种关系数据库提供了统一的访问方式，使得开发者可以编写一次程序，就能够在多种数据库上运行。它的核心功能包括建立数据库连接、执行SQL语句以及处理查询结果。

# 第三章 需求分析

需求分析是开展一个项目的前提和先决条件。通过需求分析可以很清晰了解项目用户的基本需求情况。同时，需求分析也是本项目进行开发的原因。进而对项目开发进行可行性分析，包括技术可行性、经济可行性等。以下为详细介绍。

## 3.1 技术可行性

HTML&CSS是被广泛利用的前端网页开发的技术。作为一种标记语言，网页浏览器可以阅读HTML文件，并将其渲染成可视化网页。使用HTML&CSS技术可以快速构造开发出稳定、美观的HTML前端界面。同时，HTML&CSS也具有优秀的浏览器友好性，通过结构合理的HTML有助于SEO（搜索引擎优化）。而CSS作为频繁和HTML使用的开发工具，允许HTML应用特定样式，并使样式和页面分离，能够充分满足开发时的简洁性和易懂性，并美化HTML页面。而在后端上，B/S 模式天然支持多种操作系统和设备（如 Windows、macOS 和移动端），为用户提供灵活的访问方式。系统需要支持多个用户同时访问，尤其在数据库访问和数据交互方面，应具备良好的实时性能和并发处理能力。而且JavaWeb 是基于 Java 平台开发 Web 应用程序的技术，常用的技术栈包括 Servlet、JSP 和框架（如 Spring、Struts）。Java 的“一次编写，到处运行”特性使得 JavaWeb 项目能够在多种环境中运行，如 Windows、Linux 和 macOS。此外Java提供了大量的开源框架和工具（如 Spring Boot、MyBatis），能够加快开发效率并减少重复性工作。因此，技术层面分析可行。

## 3.2 经济可行性

HTML&CSS作为网页开发的基础技术具有低廉的开发成本，与其他技术（例如原生应用开发、复杂客户端开发）相比，HTML开发网页前端无需高昂费用或专属开发工具，并且市面上有大量开源开发工具（VS Code）供开发者使用开发。除此之外，HTML技术广泛应用于网页前端开发，具有稳定性和长期支持，因此开发出的网页具有较长生命周期，更新仅涉及页面内容更新等等，降低了维护成本。并且HTML技术无论是使用受众或是开发受众都极为众多，在相关平台上，也不乏前端工程师上传的开发好的现成框架，供他人借鉴使用，进而减轻开发难度。同时，B/S 模式是一种基于浏览器和服务器的架构模式，用户通过浏览器即可访问后台服务器上的数据和应用。仅需通过浏览器即可访问系统，这种模式能够大幅降低运维和部署成本。

## 3.3 社会可行性

疫情之后，公众对健康的关注度不断增高。多元化的医疗服务也不可获取。而健康自查网站作为线上服务的重要载体之一，能够很好满足人们快捷和个性化的健康需求。除此之外，公众对于信息安全的重视也与日俱增。而Java 本身提供了完善的安全机制，加上开源社区的支持，能够有效应对 Web 应用的常见安全问题，如 SQL 注入和 XSS 攻击。不仅如此，数据库安全也是近年来的重中之重。MySQL 是目前最流行的开源关系型数据库系统之一，以其高性能、稳定性、安全性和易用性深受开发者欢迎。MySQL 支持多种查询优化技术（如索引、查询缓存），能够快速处理大量数据。 其 InnoDB 存储引擎支持事务和外键约束，确保数据的安全性和一致性。 MySQL 也支持多种操作系统，并通过标准化接口与多种语言（如 Java）无缝集成。基于此类优点，MySQL也在大量企业开发中被运用，其安全性不言而喻。因此，从社会可行性而言，该项目可行。

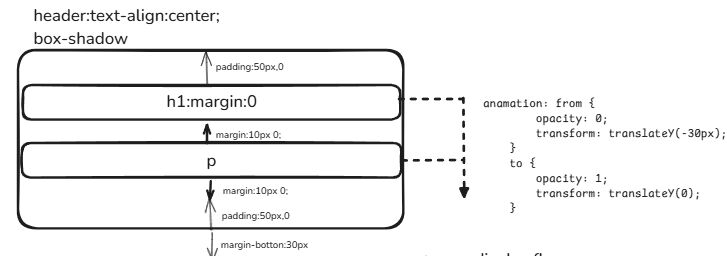
# 第四章 系统概要设计

针对系统概要设计，前端设计者应该首先使用相关设计工具，对网页框架进行设计、演示、修改。在利用工具前期设计时，前端开发人员有更多时间对自己的设计进行思考、修改，并对可能存在的设计风险进行规避，完善项目前端设计。以下为系统设计描述及结构图。

## 4.1 首页

### 4.1.1标题头设计

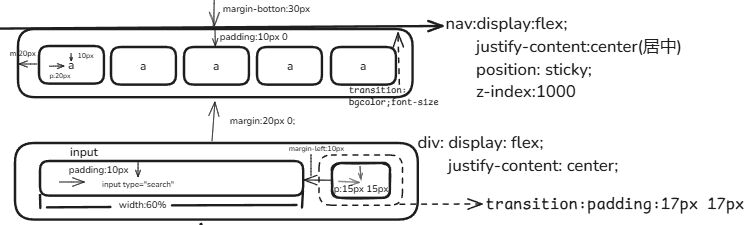
首页设计上，需要首先让用户清楚认识到，该项目名是什么，所提供的服务是什么。一方面清晰化服务内容，另一方面间接提升项目的“品牌效应”。因此利用<header>标签设计。其设计结构图如下图1所示。



**图1 header设计结构图**

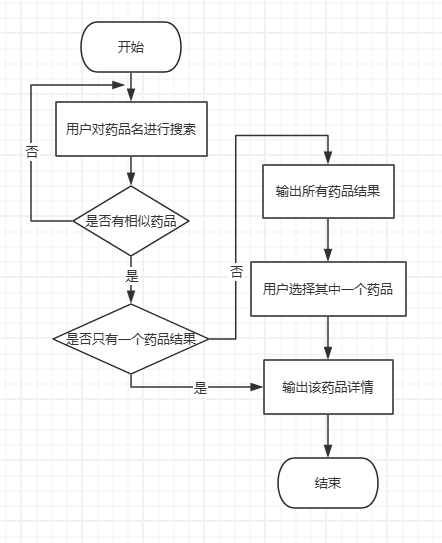
### 4.1.2定位功能设计

定位功能，即“导航栏”，对于网页设计而言具有重要的意义。在“导航栏”中用户可以清楚认识，网页有多少个模块，如何定位到自己要去的界面。此外，网页还设计了“药物搜索”功能，便于用户精确或模糊搜索，更加快速定位到自己需要的药物上。其HTML设计结构图如下图2所示。



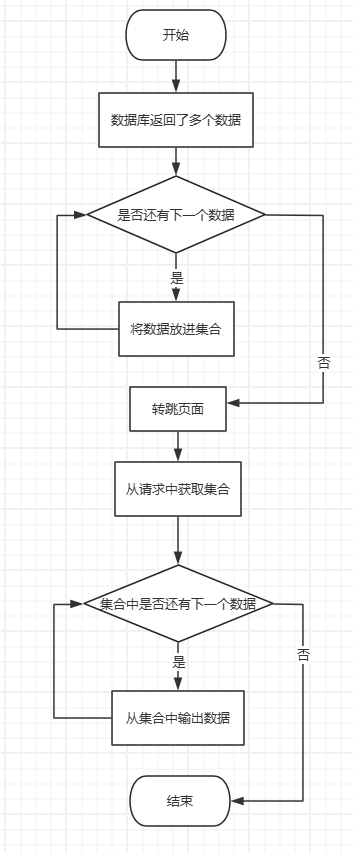
**图2 定位结构图**

用户需要键入药品的名字，搜索时，先判断数据库中是否有这个药品，如果没有则返回搜索页面；如果有则判断是否已经精确到只有一个结果。若只有一个结果，则直接展示该药品详情；若有多个结果，则在新页面中展示所有结果，供用户挑选心仪的结果，选择后展示该药品详情。后端逻辑图入下图3所示。



**图3 搜索流程图**

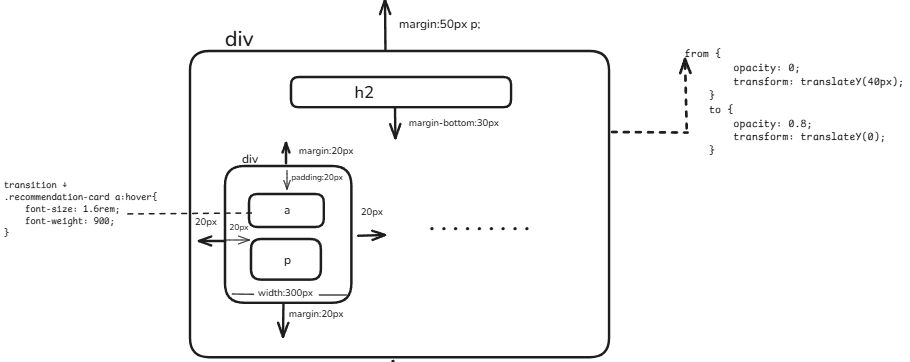
除此之外，当搜索出多个结果时，会进行对结果循环打包，在进行页面转跳时，通过请求传递给前端再次进行循环输出结果。流程图入下图4所示。



**图4 搜索多个结果流程图**

### 4.1.3“新帖速递”设计

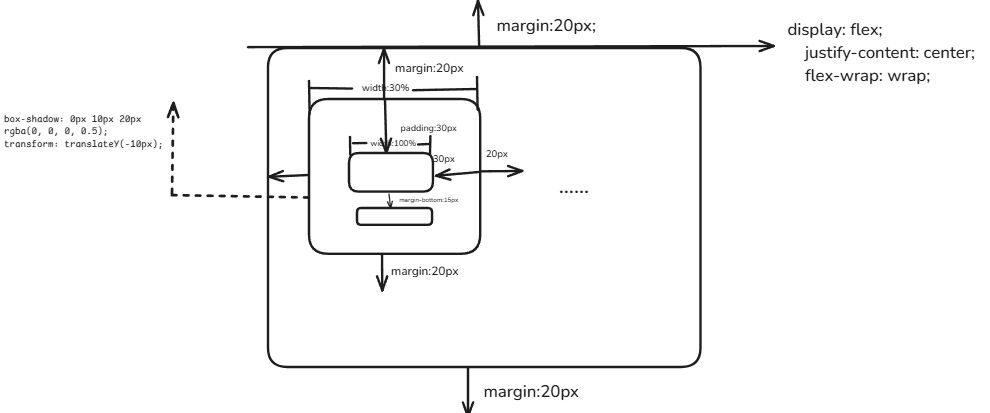
该模块涉及到前端设计的第4个模块：病情论坛。“论坛”中各类文章被更新到该板块上，推荐给用户进行阅读，增加网页使用率、点击率。设计图如下图5。



**图5 “新帖速递”结构图**

### 4.1.4导航块功能设计

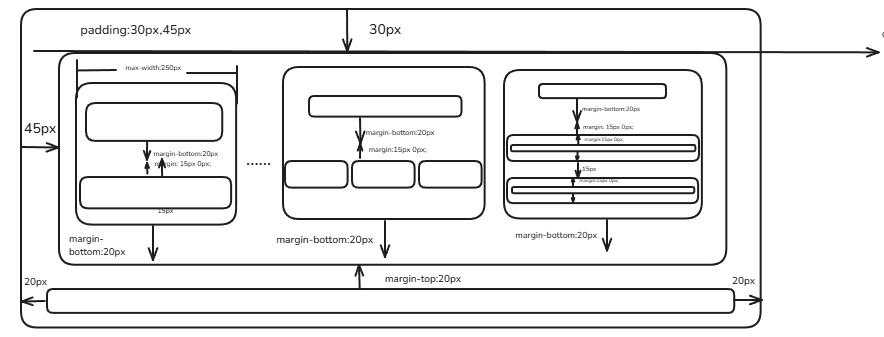
该模块与“导航栏”功能类似，然而具有图片描述和更清晰的文本描述，方便相关特殊用户，例如老年群体，快速使用。不会因为操作困难而造成使用问题。设计图如下图6。



**图6 “导航块”结构图(重复结构设计省略)**

### 4.1.5页脚设计

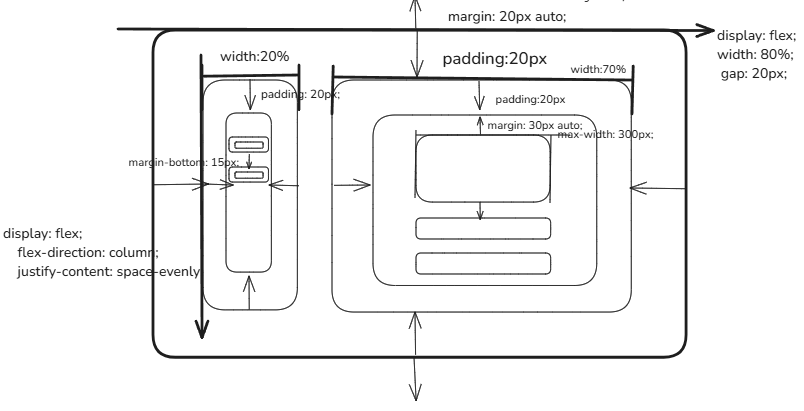
该模块作为网页底部页脚，起到“总结性”的作用。进一步提供了网页的相关信息，公开相关联系方式、社交媒体账号等等推广性内容，写出网页隐私政策等法律条文。设计图如下图7。



**图7 “导航块”结构图(重复结构设计省略)**

## 4.2 “对症下药”-药物信息一览表

新页面“对症下药”中，该页面的主要设计思路是：将药物首先分为若干个类别，并列成一个类似于导航栏的栏，供用户点击后查看不同的药物。同时，药品详细信息采用“图片-文本-文本”的方式排列，向用户阐述具体药物信息，以进行决策支持。（注：本页面标题头、页脚与首页一致。）设计结构图如下图8。



**图8 “对症下药”药品信息设计图**

## 4.3 “医院评分”界面

在该项目中，用户需要对医院进行评分，或者发送文本评论。用户可通过用户可以通过侧边栏选择不同的医院，系统应展示对应医院的评论列表。

侧边栏应包含医院名称或标识，以及可能的筛选条件（如时间范围、评论等级等）。用户可以通过侧边栏选择不同的医院，系统应展示对应医院的评论列表。侧边栏应包含医院名称或标识，以及可能的筛选条件（如时间范围、评论等级等）。用户可以通过侧边栏选择不同的医院，系统应展示对应医院的评论列表。侧边栏应包含医院名称或标识，以及可能的筛选条件（如时间范围、评论等级等）。

该界面包括输出评论功能。用户可以在选定医院的评论区域输入评论内容并提交。提交前应进行必要的验证（如非空验证、长度限制等）。提交后，评论应即时显示在评论列表中。

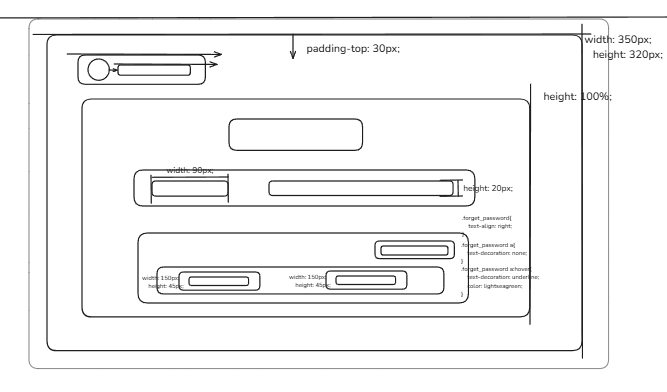
## 4.4 “病情论坛”界面

该项目中，用户可使用不同的论坛板块，查看不同的帖子，进行学习。用户可以通过侧边栏选择不同的论坛板块，系统应展示对应板块的帖子列表。侧边栏应包含板块名称或标识，以及可能的筛选条件（如最新帖子、热门帖子等）。

项目中需包含页面跳转功能，用户可以点击帖子列表中的某个帖子，系统应跳转到该帖子的详细页面。帖子详细页面应包含帖子标题、内容、是否为官方发帖等。同时，论坛帖子列表页支持切页功能，以便用户浏览更多帖子。切页功能应提供页码导航或快速跳转选项。

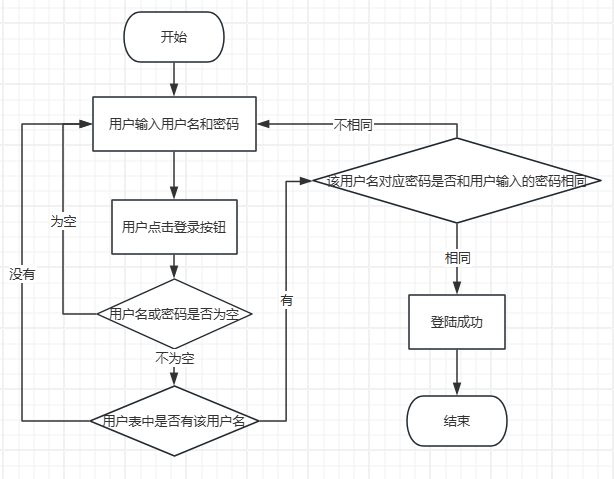
## 4.5 登录界面

在该项目中，进行相关操作时需要用户进行登录，或者注册。登录界面的编写，也普遍应用与各类网页前端设计中。其目的很简单：让用户在编写好的网页前端页面中进行用户名、密码的输入，随后继续使用功能。同时登录界面也提供了给新用户的注册界面、“忘记密码“修改界面、”返回主页“界面，可进行跳转并进行操作。设计HTML结构图如下图9。



**图9登录界面结构图**

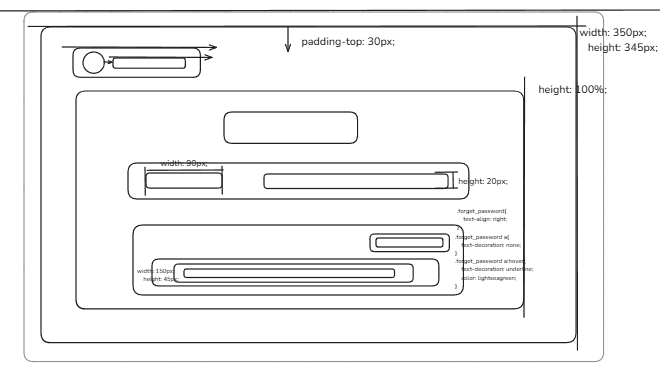
在后端操作上，系统在用户点击登录按钮后，会对用户名和密码分别进行空值判断，如果为空则返回登录页面；如果都不为空，则将在数据库中查找用户名。如果表中没有该用户则提示用户没有这个用户并返回至登录页面；如果查找到用户名，则对密码进行判断。如果密码与该用户名的密码一致则登录成功；如果不一致则提示密码不正确，并返回至登录页面。流程图入下图10所示。



**图10 登录功能流程图**

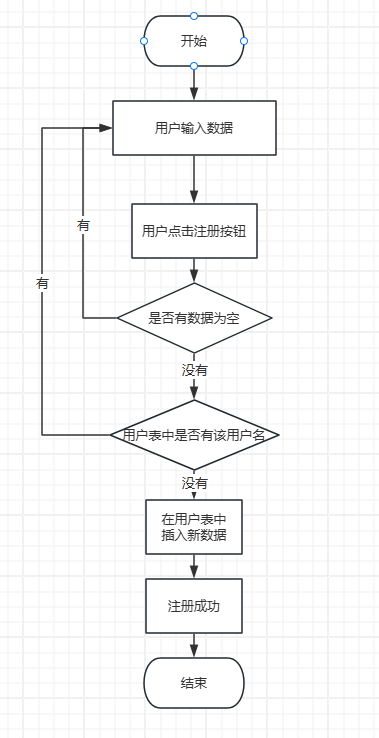
## 4.6 注册界面

注册界面与登录界面的结构上大体类似。用户通过在编写的表单中进行信息填写，即可完成注册功能。同时，相关功能也得到了保留，即：“返回首页”功能、提醒已有账户用户登录超链接。其HTML结构图如下：



**图11 注册功能**

用户点击注册按钮后，判断是否有数据未填写，若有则回到注册页面；若没有则将用户名在用户表中查找。若用户名已存在，则回到注册页面；若不存在，则在用户表中插入新数据，注册成功。流程图如下图12所示。

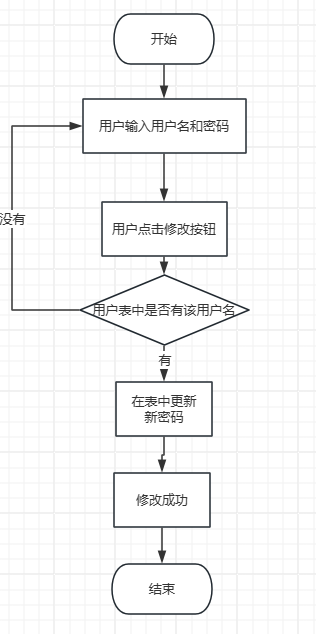


**图12 注册流程图**

## 4.7 “密码重置”界面

“密码重置”界面与前2个界面的结构上大体类似。用户通过在编写的表单中进行信息填写，即可重置密码。必要的功能也得到了保留，即：“返回首页”功能。此处HTML结构图不再赘述。

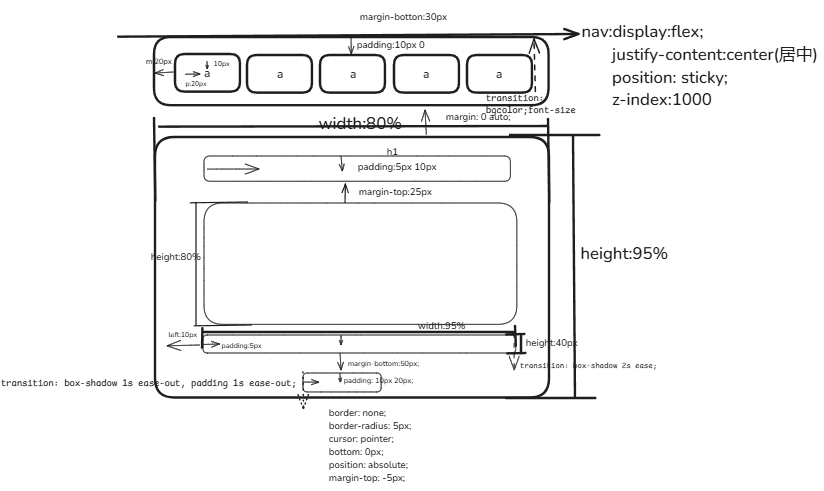
后端上，用户点击修改密码按钮以后，判断用户名和密码是否为空，如果为空则返回忘记密码页面，如果不为空则判断用户名是否存在；如果不存在则返回忘记密码页面，如果存在则将新的密码在用户表中更新，修改成功。流程图如图13所示。



**图13 重置密码流程图**

## 4.8 AI咨询界面

AI自查界面所需要设计的样式可借鉴ChatGPT的样式，即类似聊天软件一样的对话框。同时设计输入框和发送框，让用户发送信息。其结构图如下图14（此处顶部导航栏删除了标题头，但仍然保留了必要的导航栏功能。）



**图14 AI咨询界面结构图**

## 4.9 数据库结构

本项目数据库有两张表，用户表包含主键id(id)、用户名(username)、密码(password)、电话(tel)、邮箱(email)。

**表1 用户表**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **Field** | **Collation** | **Null** |
| email | varchar(64) | YES | select,insert,update,references |
| id | int | PRI | auto\_increment |
| password | varchar(64) | YES | select,insert,update,references |
| tel | varchar(11) | YES | select,insert,update,references |
| username | varchar(64) | YES | select,insert,update,references |

药品表包含主键id(id)、药品名(title)、药物样图(pic)、对应症(i)、药理(usego)。

**表2 药品表**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **Field** | **Collation** | **Null** |
| id | int | PRI | auto\_increment |
| title | varchar(45) | YES | select,insert,update,references |
| pic | varchar(255) | YES | select,insert,update,references |
| i | varchar(1000) | YES | select,insert,update,references |
| usego | varchar(1000) | YES | select,insert,update,references |

# 第五章 系统详细设计与实现

## 5.1 首页界面

首页界面是用户访问该项目的入口界面，其最主要的作用在于向用户展示项目，体现项目特色和功能一览。在首页中，用户可以使用导航栏找寻所需要的服务，利用搜索框搜寻内容，利用编写的推送帖子部分进行文章阅读，老年用户等特殊群体也能够利用“导航块”轻松找到常用功能。其界面展示图如下图5-1所示。



图5-1 网站首页

标题头HTML&CSS代码如下：

<header>

        <h1>医知便民网</h1>

        <p>您24小时的私人健康助理</p>

</header>

header{

    text-align: center;

    padding: 50px 0px;

    background-color: rgba(0, 0, 0, 0.5);

    margin-bottom: 30px;

    box-shadow: 0px 10px 20px #2c3e50;

}

@keyframes slidein {

    from {

        opacity: 0;

        transform: translateY(-30px);

    }

    to {

        opacity: 1;

        transform: translateY(0);

    }

}

header h1{

    font-size: 3.5rem;

    margin: 0;

    animation: slidein 3s, ease-out;

}

@keyframes slindeIn2 {

    from {

        opacity: 0;

        transform: translateY(-40px);

    }

    to {

        opacity: 0.8;

        transform: translateY(0);

    }

}

header p{

    font-size: 1.5rem;

    animation: slindeIn2 4s, ease-out;

    opacity: 0.8;

    margin: 10px 0 0;

}

导航栏HTML&CSS代码如下：

<nav>

<a href="../html/startSite.html">网站首页</a>

    <a href="../html/medWiki.html">对症下药</a>

    <a href="../html/rating.html">医院评分</a>

    <a href="../html/forum.html">病情论坛</a>

    <a href="../html/TaosAI/firstgptweb.html">AI自查</a>

</nav>

nav{

    display: flex;

    justify-content: center;

    padding: 10px 0;

    position: sticky;

    z-index: 1000;

}

nav a{

    color: white;

    text-decoration: none;

    padding: 10px 20px;

    margin: 0 20px;

    font-size: 1.3rem;

    border-radius: 45px;

    transition: background-color 1s ease-out, font-size 1s ease-out;

}

nav a:hover{

    background-color: rgba(255, 255, 255, 0.2);

    font-size: 1.5rem;

}

搜索框HTML&CSS代码如下：

HTML：

<div class="search-bar">

<form action="/MKCV/MedSearch" method="get">

<input type="search" placeholder="查询详细药物信息" name="key"/>

<input type="submit" value="搜索">

</form>

</div>

Css：

.search-bar{

    display: flex;

    justify-content: center;

    margin: 30px 0;

    width: 100%;

}

.search-bar form{

    width: 100%;

    max-width: 800px;

    display: flex;

    justify-content: space-between;

    align-items: center;

}

.search-bar input[type="search"]{

    padding: 10px;

    width: 100%;

    height: 100%;

    border-radius: 50px;

    outline:  rgba(255, 255, 255, 0.2);

    border: none;

    font-size: 1.1rem;

    transition: box-shadow 2s ease;

}

.search-bar input[type="search"]:focus{

    box-shadow: 0px 20px 15px rgba(255, 255, 255, 0.2);

}

.search-bar input[type="submit"]{

    margin-left: 10px;

    padding: 15px 15px;

    border-radius: 15px;

    border: none;

    outline:  rgba(255, 255, 255, 0.2);

    font-size: 1rem;

    background-color: #5B61F9;

    color: white;

    box-shadow: 0 10px 10px #353CF6;

    cursor: pointer;

    font-weight: bolder;

    transition: padding 1.2s ease, background-color 1.2s ease;

}

.search-bar input[type="submit"]:hover{

    padding: 17px 17px;

    background-color: #00c6ff;

}

服务器接收到请求后，会对地址进行解释，最终通过注解找到Servlet中的MedSearch.java并把请求转发给它。

当这个类接收到get请求后，自动触发doGet方法，执行其中的代码。

先从请求中获取搜索框name的容器中的数据，再判断其是否为空：如果为空则代表用户没输入搜索内容，在响应中写入JavaScript代码，触发后将提醒用户，并且回到刚刚的搜索页面；如果不为空，将先实例化dao层中的MedDB对象，调用里面的方法，在数据库中的药品表找相似的药品名。

当没找到时，直接响应JavaScript代码，返回搜索页面；当有结果后，先判断是否是多个结果：如果结果只有一条，则把搜索到的药品名、药品样图等信息存放进实例化好的Med对象中，通过请求转发到药品详情界面；如果为真，将会把结果中的每一条数据，通过循环，全都打包进list集合中，再讲list放进请求的标签”list”中，并把请求转跳到新的页面。新的页面也通过循环，将每一条结果一一输出，供用户选择。代码如下：

@WebServlet("/MedSearch")

public class MedSearch extends HttpServlet {

private static final long serialVersionUID = 1L;

DbUtil dbUtil=new DbUtil();

public MedSearch() {

super();

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String keyString=request.getParameter("key");

response.setContentType("text/html;charset=utf-8");

if (keyString.equals("")) {

response.getWriter().print("<script language='javascript'>alert('请在搜索框中输入内容');window.history.back();</script>");

}else {

MedDB medDB=new MedDB();

try {

ResultSet rs=medDB.name(dbUtil.getConnection(), keyString);

if (rs.last()) {

if (rs.getRow()>1) {

rs.first();

List<String> list=new ArrayList<String>();

while (rs.next()) {

list.add(rs.getString("title"));

}

request.setAttribute("list", list);

String url ="/jsp/fussySearch.jsp";

request.setCharacterEncoding("utf-8");

getServletContext().getRequestDispatcher(url).forward(request, response);

}else {

Med med=new Med();

med.setPic(rs.getString("pic"));

med.setTitle(rs.getString("title"));

med.setUsego(rs.getString("usego"));

med.setI(rs.getString("i"));

request.setAttribute("med", med);

String url ="/jsp/searchMediResult.jsp";

request.setCharacterEncoding("utf-8");

getServletContext().getRequestDispatcher(url).forward(request, response);

}

}else {

response.getWriter().print("<script language='javascript'>alert('没查询到你想要的的相关药品');window.history.back();</script>");

}

} catch (Exception e) {

e.printStackTrace();

response.getWriter().print("<script language='javascript'>alert('没查询到你想要的的相关药品');window.history.back();</script>");

}

}}}

“推荐区域”HTML&CSS代码如下：

<div class="recommendations">

        <h2>新帖速递</h2>

        <div class="recommendation-card">

            <a href="../html/forum.html">生活中的健康</a>

            <p>了解日常饮食、运动和心理健康的重要性。</p>

        </div>

        <div class="recommendation-card">

            <a href="../html/medWiki.html">如何选择药物</a>

            <p>如何正确选择和使用药品，避免不良反应。</p>

        </div>

        <div class="recommendation-card">

            <a href="../html/forum.html">今日医学突破</a>

            <p>关注最新的医学研究成果，了解未来的健康趋势。</p>

        </div>

    </div>

@keyframes slideIn3{

    from {

        opacity: 0;

        transform: translateY(40px);

    }

    to {

        opacity: 0.8;

        transform: translateY(0);

    }

}

.recommendations {

    text-align: center;

    margin: 50px 0;

    animation: slideIn3 2s ease;

}

.recommendations h2 {

    margin-bottom: 30px;

    font-size: 2rem;

    font-weight: bolder;

    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.recommendation-card {

    background-color: rgba(255, 255, 255, 0.1);

    border-radius: 10px;

    box-shadow: 0px 10px 20px rgba(0, 0, 0, 0.5);

    margin: 20px;

    padding: 20px;

    text-align: left;

    width: 300px;

    display: inline-block;

}

.recommendation-card a{

    text-decoration: none;

    margin-top: 0;

    font-size: 1.5rem;

    color: white;

    font-weight: 700;

    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

    transition: font-size 1s ease, font-weight 1s ease;

}

.recommendation-card a:hover{

    font-size: 1.6rem;

    font-weight: 900;

}

“导航块”HTML&CSS代码：

<div class="container">

        <a href="../html/medWiki.html" class="content">

            <img src="../img/startSite-Med.jpg">

            <p>学习对症下药</p>

        </a>

        <a href="../html/rating.html" class="content">

            <img src="../img/startSite-snake.jpg">

            <p>选择最佳医院</p>

        </a>

        <a href="../html/forum.html" class="content">

            <img src="../img/startSite-club.jpg">

            <p>与他人分享你的经验</p>

        </a>

        <a href="../html/TaosAI/firstgptweb.html" class="content">

            <img src="../img/startSite-AI.jpg">

            <p>与我们的AI沟通</p>

        </a>

    </div>

.container{

    display: flex;

    justify-content: center;

    flex-wrap: wrap;

    margin: 20px;

}

.content{

    margin: 20px;

    padding: 30px;

    background-color: #00c6ff;

    border-radius: 45px;

    color: white;

    text-align: center;

    text-decoration: none;

    width: 30%;

    transition: box-shadow 1s ease,transform 1s ease;

}

.content:hover{

    box-shadow: 0px 10px 20px rgba(0, 0, 0, 0.5);

    transform: translateY(-10px);

}

.content img {

    width: 100%;

    height: 300px;

    border-radius: 10px;

    margin-bottom: 15px;

    object-fit: cover;  /\* 保持比例，裁剪图片 \*/

}

.content p {

    font-size: 1.2rem;

    font-weight: bold;

    color: #fff;

}

页脚HTML&CSS代码如下：

<footer>

        <div class="footer-container">

            <div class="footer-section">

                <h4>关于我们</h4>

                <p>"医知便民"网致力于为用户提供全面、可靠的医药信息与便捷的健康服务。<br>通过整合专业医疗知识和实用资源，我们为公众提供药物查询、健康指导等功能，帮助大家在日常生活中做出明智的健康决策。</p>

            </div>

            <div class="footer-section">

                <h4>联系我们</h4>

                <p>手机号: +86 15162537431</p>

                <p>邮箱: MedfuerAlle@outlook.com</p>

                <p>地址: 江苏省无锡市新吴区</p>

            </div>

            <div class="footer-section">

                <h4>社交媒体</h4>

                <a href="https://x.com/"><img src="../img/icon\_x.png"></a>

                <a href="https://m.weibo.cn/"><img src="../img/icon\_weibo.png"></a>

                <a href="https://www.linkedin.com/"><img src="../img/icon\_linkin.png"></a>

            </div>

            <div class="footer-section">

                <h4>本站政策</h4>

                <p><a href="../html/privacy.html">隐私政策</a></p>

                <p><a href="#">使用条款</a></p>

                <p><a href="#">Cookie 政策</a></p>

            </div>

        </div>

        <p>&copy; 所有的数据来源于Internet</p>

    </footer>

footer {

    text-align: center;

    padding: 30px 45px;

    background-color: rgba(0, 0, 0, 0.5);

    color: #fff;

}

.footer-container{

    display: flex;

    justify-content: space-between;

    flex-wrap: wrap;

}

.footer-section{

    margin-bottom: 20px;

    max-width: 250px;

    margin-right: 75px;

}

.footer-section h4{

    font-size: 1.3rem;

    margin-bottom: 20px;

    color: #ddd;

    font-weight:800;

    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

}

.footer-section p, .footer-section a {

    color: #ddd;

    font-size: 0.9rem;

    margin: 15px 0px;

    text-decoration: none;

}

.footer-section a:hover{

    text-decoration: underline;

    color: #00c6ff;

}

.footer-section a img{    /\*"社交媒体"icon\*/

    width: 30px;

    height: 30px;

    margin: 20px;

}

footer p{

    text-align: center;

    margin-top: 20px;

    margin-left: 20px;

    margin-right: 20px;

    font-size: 0.8rem;

    font-family:'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif

}

## 5.2 “对症下药”界面

用户可以查看前端设计中所有的药物详细信息。并通过药品分类栏，查看各个种类的相关药品。其界面如下图5-2所示。



**图5-2 “对症下药”**

## 5.3 医院评分

用户进入此界面后可以通过左边的侧边栏选择合适的医院，点击后即可跳转，其界面展示如下图5-3所示。



**图5-3 评分选择界面**

跳转页面后用户可以查看各医院的评论，也可以留下自己的评论，其界面展示如下图5-4所示。



**图5-4 医院评分评论界面**

跳转代码和医院评论Javascript代码如下：

document.addEventListener('DOMContentLoaded', function() {

    const hospitalInfo = document.getElementById('hospital-info');

    const commentsSection = document.getElementById('comments-section');

    const sidebarLinks = document.querySelectorAll('.sidebar a');

    let isCommentSubmitted = false;

    let userRating = 0;

    let userCommentText = '';

    sidebarLinks.forEach(link => {

        link.addEventListener('click', function(event) {

            event.preventDefault();

            const hospital = this.getAttribute('data-hospital');

            updateHospitalInfo(hospital);

            updateComments(hospital);

        });

    });

    function updateHospitalInfo(hospital) {

        hospitalInfo.innerHTML = '';

        const hospitalData = {

            'A': {

                title: '医院 A',

                meta: '位置: 城市 X',

                description: '这是医院 A 的简要介绍...',

                rating: 4

            },

            'B': {

                title: '医院 B',

                meta: '位置: 城市 Y',

                description: '这是医院 B 的简要介绍...',

                rating: 3

          },

            'C': {

                title: '医院 C',

                meta: '位置: 城市 Z',

                description: '这是医院 C 的简要介绍...',

                rating: 5

            },

            'D': {

                title: '医院 D',

                meta: '位置: 城市 W',

                description: '这是医院 D 的简要介绍...',

                rating: 2

            }

        };

        if (hospitalData[hospital]) {

            const data = hospitalData[hospital];

            hospitalInfo.innerHTML = `

                <div class="hospital">

                    <h3 class="hospital-title">${data.title}</h3>

                    <p class="hospital-meta">${data.meta}</p>

                    <p>${data.description}</p>

                    <div class="hospital-rating">

                        <div class="rating-stars read-only">

                            ${[1, 2, 3, 4, 5].map(i => `

                                <label data-value="${i}" class="${i <= data.rating ? 'active' : ''}">&#9733;</label>

                            `).join('')}

                        </div>

                    </div>

                </div>

            `;

        } else {

            hospitalInfo.innerHTML = '<p>请选择医院</p>';

        }

    }

    function updateComments(hospital) {

        const comments = {

            'A': [

                { user: '用户1', date: '2024年9月1日', text: '医院 A 的服务非常好，医生很专业。' },

                { user: '用户2', date: '2024年9月2日', text: '医院 A 的环境不错。' }

            ],

            'B': [

                { user: '用户3', date: '2024年9月3日', text: '医院 B 的环境不错，但等候时间较长。' }

            ],

            'C': [

                { user: '用户4', date: '2024年9月4日', text: '医院 C 的医生很细心，环境也很舒适。' }

            ],

            'D': [

                { user: '用户5', date: '2024年9月5日', text: '医院 D 的服务一般，等待时间较长。' }

            ]

        };

        const hospitalRatings = {

            'A': 4,

            'B': 3,

            'C': 5,

            'D': 2

        };

        if (comments[hospital]) {

            const commentsHtml = comments[hospital].map(comment => `

                <div class="comment">

                    <p class="comment-meta">${comment.user} - ${comment.date}</p>

                    <p>${comment.text}</p>

                    <div class="comment-fixed-rating">

                        ${[1, 2, 3, 4, 5].map(i => `

                            <label class="${i <= hospitalRatings[hospital] ? 'active' : ''}">&#9733;</label>

                        `).join('')}

                    </div>

                </div>

            `).join('');

            commentsSection.innerHTML = `

                <h3>评论</h3>

                ${commentsHtml}

                <div class="comment-form ${isCommentSubmitted ? 'read-only' : ''}">

                    <h4>添加评论</h4>

                    <div class="rating-stars">

                        <label data-value="1">&#9733;</label>

                        <label data-value="2">&#9733;</label>

                        <label data-value="3">&#9733;</label>

                        <label data-value="4">&#9733;</label>

                        <label data-value="5">&#9733;</label>

                    </div>

                    <textarea rows="4" placeholder="在此输入您的评论..." ${isCommentSubmitted ? 'disabled' : ''}></textarea>

                    <button id="submit-comment">${isCommentSubmitted ? '您已提交评论' : '提交评论'}</button>

                    <p id="submission-message" class="submission-success" style="display:none;">评论提交成功!</p>

                </div>

            `;

            const ratingStars = document.querySelectorAll('.comment-form .rating-stars label');

            ratingStars.forEach(star => {

                star.addEventListener('click', function() {

                    if (!isCommentSubmitted) {

                        userRating = this.getAttribute('data-value');

                        ratingStars.forEach(s => {

                            s.classList.remove('active');

                            if (parseInt(s.getAttribute('data-value')) <= userRating) {

                                s.classList.add('active');

                            }

                        });

                    }

                });

            });

            document.querySelector('.comment-form textarea').addEventListener('input', function() {

                userCommentText = this.value;

            });

            document.getElementById('submit-comment').addEventListener('click', function() {

                if (!isCommentSubmitted) {

                    const commentForm = document.querySelector('.comment-form');

                    const submissionMessage = document.getElementById('submission-message');

                    commentForm.querySelector('textarea').disabled = true;

                    commentForm.querySelectorAll('.rating-stars label').forEach(label => {

                        label.classList.remove('active');

                        label.removeEventListener('click', null);

                    });

                    submissionMessage.style.display = 'block';

                    const commentFixedRating = document.createElement('div');

                    commentFixedRating.classList.add('comment-fixed-rating');

                    commentFixedRating.innerHTML = `

                        ${[1, 2, 3, 4, 5].map(i => `

                            <label class="${i <= userRating ? 'active' : ''}">&#9733;</label>

                        `).join('')}

                    `;

                    commentsSection.innerHTML += `

                        <div class="comment">

                            <p class="comment-meta">您 - 当前时间</p>

                            <p>您的评论内容</p>

                            <p>${userCommentText}</p>

                        </div>

                    `;

                    commentsSection.querySelector('.comment:last-child').appendChild(commentFixedRating);

                    this.textContent = '您已提交评论';

                    isCommentSubmitted = true;

                } else {

                    this.textContent = '您已提交评论';

                }

            });

        } else {

            commentsSection.innerHTML = '';

        }

    }

});

## 5.4 病情论坛

用户进入此界面后，可通过左边的侧边栏，选择相关板块界面，点击后即可跳转到该板块，同时对应的帖子也可以被预览。其展示界面如下图5-5以及5-6。



**图5-5 论坛内容界面**



**图5-6 切换模块**

用户在论坛界面时，也可以通过帖子预览去底部的分页按钮，跳转到第二页，浏览更多的文章。界面展示如下图5-7所示。



**图5-7 分页功能**

分页功能具体代码如下：

        document.addEventListener('DOMContentLoaded', () => {

            const postsPerPage = 4;

            const posts = document.querySelectorAll('.post');

            const totalPosts = posts.length;

            const totalPages = Math.ceil(totalPosts / postsPerPage);

            let currentPage = 1;

            const createPaginationButtons = () => {

                const pagination = document.getElementById('pagination');

                pagination.innerHTML = '';

                if (totalPages <= 1) return;

                for (let i = 1; i <= totalPages; i++) {

                    const button = document.createElement('button');

                    button.textContent = i;

                    button.classList.add('page-btn');

                    if (i === currentPage) {

                        button.classList.add('active');

                    }

                    button.addEventListener('click', () => {

                        currentPage = i;

                        showPage(currentPage);

                    });

                    pagination.appendChild(button);

                }

            };

            const showPage = (page) => {

                posts.forEach((post, index) => {

                    post.style.display = (index >= (page - 1) \* postsPerPage && index < page \* postsPerPage) ? 'block' : 'none';

                });

                const buttons = document.querySelectorAll('.pagination button');

                buttons.forEach(button => {

                    button.classList.toggle('active', parseInt(button.textContent) === page);

                });

            };

            createPaginationButtons();

            showPage(currentPage);

        });

用户可以点击自己所感兴趣的文章，即可查看文章全部内容。除此之外，右上角会显示是官方发帖，还是网友发贴，体现信息来源。其界面展示如下图5-8所示。



**图5-8浏览帖子功能**

## 5.5 AI自查

用户可以在自查界面与AI进行对话。通过在输入框中输入文本，并通过设计的发送按钮进行发送，获取AI回答。其界面如下图5-9所示。



**图5-9 AI自查界面**

导航栏HTML&CSS代码如下：

<nav>

        <a href="../../html/startSite.html">网站首页</a>

        <a href="../../html/medWiki.html">对症下药</a>

        <a href="../../html/rating.html">医院评分</a>

        <a href="../../html/forum.html">病情论坛</a>

        <a href="../../html/TaosAI/firstgptweb.html" style="background: grey;">AI自查</a>

</nav>

nav{

    display: flex;

    justify-content: center;

    padding: 10px 0;

    position: sticky;

    z-index: 1000;

}

nav a{

    color: white;

    text-decoration: none;

    padding: 10px 20px;

    margin: 0 20px;

    font-size: 1.3rem;

    border-radius: 45px;

    transition: background-color 1s ease-out, font-size 1s ease-out;

}

nav a:hover{

    background-color: rgba(255, 255, 255, 0.2);

    font-size: 1.5rem;

}

聊天界面HTML&CSS代码如下：

<div class="container">

        <h1>医知助手</h1>

        <div id="output-text">

            <dl id="theList">

            </dl>

        </div>

        <textarea id="input-text" placeholder="请输入要查询的健康问题"></textarea>

        <button id="submit-button">发送</button>

        <script src="../../js/TaosAI/script.js"></script>

    </div>

.container {

    width: 80%;

    height: 95%;

    margin: 0 auto;

    text-align: center;

    background-color: #ffffff;

    position: relative;

}

.container h1{

    color: white;

    padding: 5px 10px;

    font-weight: bolder;

    background: linear-gradient(135deg,#00c6ff,#0072ff);

}

#output-text {

    height: 80%;

    margin-top: 25px;

    text-align: left;

    overflow-y:auto;

}

dt {

    font-weight: bold;

    padding: 10px;

}

textarea {

    width: 95%;

    height: 40px;

    resize: none;

    padding: 5px;

    margin-bottom: 50px;

    left: 10px;

    bottom: 0px;

    position: absolute;

    border-radius: 10px;

    border-color: #87B2F6;

    border-width: 3px;

    transition: box-shadow 2s ease;

}

textarea:focus{

    box-shadow: 0px 20px 15px darkgrey;

}

textarea::placeholder{

    font-size: 1rem;

    font-weight: bolder;

}

button {

    padding: 10px 20px;

    background-color: #007bff;

    color: white;

    border: none;

    border-radius: 5px;

    cursor: pointer;

    bottom: 0px;

    position: absolute;

    margin-top: -5px;

    transition: box-shadow 1s ease-out, padding 1s ease-out;

}

button:hover{

    box-shadow: 0px 10px 15px rgba(0, 0, 0, 0.3);

    padding: 12px 22px;

}

img {

  border-radius: 50%;

}

## 5.6 登录界面

登录界面用于用户进行个人账号登录。用户可根据设计的表单填写自己的用户名及密码。登录界面中确保了功能的完整性，即具有新用户注册跳转链接、忘记密码重置链接、返回首页链接。其界面如下图5-10所示。



**图5-10登录界面**

登录界面全局HTML&CSS代码如下：

HTML：

@WebServlet("/LoginServlet")

public class LoginServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

String message="";

public LoginServlet() {

super();

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

request.setCharacterEncoding("utf-8");

response.setContentType("text/html;charset=utf-8");

String username = request.getParameter("username");

String password = request.getParameter("password");

if (username.equals("")) {

response.getWriter().print("<script language='javascript'>alert('请输入用户名');window.history.back();</script>");

}else if (password.equals("")) {

response.getWriter().print("<script language='javascript'>alert('请输入密码');window.history.back();</script>");

}

else {

User user = UserDB.selectUser(username);

if(user != null) {

if(user.getPassword().equals(password)) {

response.getWriter().print("<script language='javascript'>alert('登录成功');window.location.href='/MKCV/jsp/rating.jsp';</script>");

}

else {

message = "密码不正确！";

response.getWriter().print("<script language='javascript'>alert('密码不正确！');window.location.href='/MKCV/jsp/login.jsp';</script>");

}

}

else {

message = "用户不存在！";

response.getWriter().print("<script language='javascript'>alert('用户不存在！');window.location.href='/MKCV/jsp/login.jsp';</script>");

}

}

}

}

CSS：

@keyframes gradientTransition {

    0% {

        background-position: 0% 50%;

    }

    50% {

        background-position: 100% 50%;

    }

    100% {

        background-position: 0% 100%;

    }

}

body {

    margin: 0;

    overflow-x: hidden;

    font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva, Verdana, sans-serif;

    background: linear-gradient(to right, #FF9A8B 20%, #FF6A88 50%, #FF99AC 80%),

    linear-gradient(90deg, #FBDA61 0%, #FF5ACD 100%);

    background-size: 300% 300%;

    background-blend-mode: soft-light;

    animation: gradientTransition 14s ease-in-out infinite;

    color: black;

    height: 100vh;

}

.main\_body{

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

}

.login\_body{

    width: 350px;

    height: 320px;

    padding-top: 30px;

    padding: 50px;

    border-radius: 10px;

    background-color: rgba(255, 255, 255, 0.8);

}

.back\_box {

    display: flex;

    align-items: center;

}

.back\_box a{

    display: flex;

    align-items: center;

    text-decoration: none;

    color: #333;

    font-size: 16px;

    font-weight: bold;

    transition: color 0.3s ease;

}

.back\_box a:hover {

    color: lightseagreen;

}

.back\_box img{

    width: 20px;

    height: 20px;

    margin-right: 8px;

    transition: transform 0.3s ease;

}

.back\_box a:hover img {

    transform: scale(1.1);

}

form{

    height: 100%;

    display:flex;

    flex-direction: column;

    justify-content: space-between;

    align-items: center;

}

h2{

    font-size: 34px;

}

span{

    font-size: 18px;

    font-weight: bold;

    display: inline-block;

    width: 90px;

    text-align: left;

}

input{

    height: 20px;

    border: none;

    transition: box-shadow 2s ease;

    border-radius: 10px;

    font-weight: bold;

}

input:focus{

    box-shadow: 0px 20px 15px darkgrey;

}

.btn\_box\_son{

    padding: 10px;

    margin-top: 10px;

}

.btn\_box\_son .login{

    margin-right: 5px;

}

.btn\_box\_son .regis{

    margin-left: 5px;

}

.forget\_password{

    text-align: right;

}

.forget\_password a{

    text-decoration: none;

}

.forget\_password a:hover{

    text-decoration: underline;

    color: lightseagreen;

}

.login{

    border: none;

    width: 150px;

    height: 45px;

    border-radius: 10px;

    background-color: lightskyblue;

    color: black;

    font-size: 18px;

    font-weight: bold;

}

.login:hover{

    cursor: pointer;

    box-shadow: 0 20px 15px darkcyan;

}

.regis{

    border: none;

    width: 150px;

    height: 45px;

    border-radius: 10px;

    background-color: lightskyblue;

    color: black;

    font-size: 18px;

    font-weight: bold;

}

.regis:hover{

    cursor: pointer;

    box-shadow: 0 20px 15px darkcyan;

}

当用户点击当按钮后会触发form表单的action，向服务器的“/MKCV/LoginServlet”地址发送post请求。Post请求在传递值时不会显示在浏览器的地址栏，更加安全更加合理。

Get请求后会直接跟传递的值



Post请求则没有



当LoginServlet类接收到post请求后，自动触发doPost方法，执行其中的代码。

先从请求中获取用户名输入框username和密码输入框password容器中的数据，再先后判断是否为空：如果为空则代表没输入对应内容，在响应中写入JavaScript代码，返回触发后将提醒用户，并且回到登录页面；如果全都不为空，将先实例化dao层中的UserDB对象，调用里面的方法，查找数据库中的用户表，是否有对应的用户名，如果没有则返回登录页面，如果有则判断密码是否一致。不一致则返回，一致则登陆成功。

@WebServlet("/LoginServlet")//读取/LoginServlet完整路径

public class LoginServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

String message="";//定义一个变量

public LoginServlet() {

super();

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

//设置编码

request.setCharacterEncoding("utf-8");

response.setContentType("text/html;charset=utf-8");

//获取登入的信息(姓名和密码)

String username = request.getParameter("username");

String password = request.getParameter("password");

if (username.equals("")) {

response.getWriter().print("<script language='javascript'>alert('请输入用户名');window.history.back();</script>");

}else if (password.equals("")) {

response.getWriter().print("<script language='javascript'>alert('请输入密码');window.history.back();</script>");

}

else {

User user = UserDB.selectUser(username);

if(user != null) {

if(user.getPassword().equals(password)) {

response.getWriter().print("<script language='javascript'>alert('登录成功');window.location.href='/MKCV/jsp/rating.jsp';</script>");

}

else {

message = "密码不正确！";

response.getWriter().print("<script language='javascript'>alert('密码不正确！');window.location.href='/MKCV/jsp/login.jsp';</script>");

}

}

else {

message = "用户不存在！";

response.getWriter().print("<script language='javascript'>alert('用户不存在！');window.location.href='/MKCV/jsp/login.jsp';</script>");

}

}

}

}

## 5.7 注册界面

注册界面，即用户通过输入用户名、邮箱、密码等信息进行注册。设计风格和模式与登录界面保持统一。同时相应功能依旧有保留：对有账号用户的提示登录跳转链接、返回首页链接。其界面如图5-11所示。



**图5-11 注册界面**

当RegisterServlet类接收到post请求后，自动触发doPost方法，执行其中的代码。

先从请求中获取用户名输入框username、邮箱输入框email、电话输入框number、密码输入框password和确认密码输入框repassword容器中的数据，再一一判断是否为空：如果任意为空则代表没输入对应内容，在响应中写入JavaScript代码，返回触发后将提醒用户，并且回到登录页面；如果全都不为空，将判断密码和重复密码是否一致，再调用UserDB中的方法，判断用户名是否存在，如不存在则实例化pojo层中的User对象，把所有数据都封装进去，再调用UserDB里的insert方法，将封装好的User插入到用户表中，注册成功。

@WebServlet("/RegisterServlet")

public class RegisterServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

String message ="";

public RegisterServlet() {

super();

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

request.setCharacterEncoding("utf-8");

response.setContentType("text/html;charset=utf-8");

String username = request.getParameter("username");

String password = request.getParameter("password");

String password2 = request.getParameter("repassword");

String tel=request.getParameter("number");

String email = request.getParameter("email");

if(username.equals("")){

response.getWriter().print("<script language='javascript'>alert('请输入用户名');window.history.back();</script>");

}else if(password.equals("")){response.getWriter().print("<script language='javascript'>alert('请输入密码');window.history.back();</script>");}

else if (password2.equals("")) {response.getWriter().print("<script language='javascript'>alert('请输入重复密码');window.history.back();</script>");}

else if (email.equals("")) {response.getWriter().print("<script language='javascript'>alert('请输入邮箱');window.history.back();</script>");}

else if (tel==null) {response.getWriter().print("<script language='javascript'>alert('请输入电话');window.history.back();</script>");}

else {User user = new User();

user.setUsername (username);

user.setPassword(password);

user.setEmail(email);

user.setTel(tel);

if(password.equals(password2))

{

if( !UserDB.UserExists(username) ) {

UserDB.insert(user);

response.getWriter().print("<script language='javascript'>alert('注册成功');window.location.href='/MKCV/jsp/login.jsp';</script>");

}

else

{

response.setContentType("text/html;charset=utf-8");

try {

response.getWriter().print("<script language='javascript'>alert('该用户名已被占用');window.location.href='/MKCV/jsp/register.jsp';</script>");

} catch (IOException e) {

e.printStackTrace();

}

}

}

else

{

response.setContentType("text/html;charset=utf-8");

try {

response.getWriter().print("<script language='javascript'>alert('两次密码不一致');window.location.href='/MKCV/jsp/register.jsp';</script>");

} catch (IOException e) {

// TODO 自动生成的 catch 块

e.printStackTrace();

}

}}}}

## 5.8密码重置界面

密码重置界面，即在用户忘记密码时提供密码修改的界面。设计风格与登录、注册一致。并且通过提供相关个人信息，进行密码的重置。同时，“返回首页“功能依旧保持。其界面图如图5-12所示。



**图5-12密码重置**

当用户点击重置按钮后会触发form表单的action，向服务器的“/MKCV/ResetPasswordServlet”地址发送post请求。当ResetPasswordServlet类接收到post请求后，自动触发doPost方法，执行其中的代码。

先从请求中获取用户名输入框username、密码输入框password和确认密码输入框repassword容器中的数据，再一一判断是否为空：如果任意为空则代表没输入对应内容，在响应中写入JavaScript代码，返回触发后将提醒用户，并且回到登录页面；如果全都不为空，将判断两次密码是否一致，再调用UserDB中的方法，判断用户名是否存在。如不存在，将返回忘记密码页面；如果存在则实例化pojo层中的User对象，把所有数据都封装进去，再调用UserDB里的update方法，将封装好的User更新username更新到用户表中，修改密码成功。代码如下：

@WebServlet("/ResetPasswordServlet")

public class ResetPasswordServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

request.setCharacterEncoding("utf-8");

response.setContentType("text/html;charset=utf-8");

String username = request.getParameter("username");

String password = request.getParameter("password");

String password2 = request.getParameter("repassword");

if (username.equals("")) {response.getWriter().print("<script language='javascript'>alert('请输入用户名');window.history.back();</script>");}

else if (password.equals("")) {response.getWriter().print("<script language='javascript'>alert('请输入密码');window.history.back();</script>");}

else if (password2.equals("")) {response.getWriter().print("<script language='javascript'>alert('请输入重复密码');window.history.back();</script>");}

else {if(password.equals(password2))

{

if( UserDB.UserExists(username) ) {

User user = new User();

user.setUsername (username);

user.setPassword(password);

UserDB.update(user);

response.getWriter().print("<script language='javascript'>alert('更新密码成功！');window.location.href='/MKCV/jsp/login.jsp';</script>");

}

else

{

try {

response.getWriter().print("<script language='javascript'>alert('该用户名不存在');window.history.back();</script>");

} catch (IOException e) {

// TODO 自动生成的 catch 块

e.printStackTrace();

}}}

else

{

try {

response.getWriter().print("<script language='javascript'>alert('两次密码不一致');window.history.back();</script>");

} catch (IOException e) {

// TODO 自动生成的 catch 块

e.printStackTrace();

}}}}}

# 第六章 系统测试与维护

## 6.1 测试目的

系统测试旨在确保各功能模块的完整性、操作便利性及设计合理性，并验证其是否符合预期效果和用户需求。通过测试，能够发现设计中潜在的问题，便于及时修复和优化，从而减少日后可能出现的故障风险。此外，测试过程中往往会涌现新的思路，为进一步提升系统设计提供依据。通过前后端紧密配合，系统不仅能够成功实现预期功能，还在测试中进一步优化了性能和稳定性，为用户提供了良好的交互体验。

## 6.2 测试结果

### 6.2.1前端

1.首页设计

成功实现首页基本布局，用户可清晰导航至“对症下药”、AI自查、登录、注册等模块，界面简洁，操作方便。

2.搜索框

输入精准药品名时，跳转至药品详情页面，样式显示正常，信息分布合理；

输入模糊关键词时，跳转至模糊结果页面，界面显示效果无误。

3.侧边栏跳转

验证侧边栏中的医院列表是否正确显示，点击不同医院是否能正确切换并显示对应医院的评论列表。

4.评论功能

验证评论输入框是否对非法字符、空内容等进行有效验证。

测试评论提交功能，确保评论能正确保存到数据库并即时显示在评论列表中。

5.分页逻辑

验证帖子列表的分页逻辑是否正确，确保不同页码下的帖子数据正确显示。

### 6.2.2 后端

1.登录功能

用户名或密码为空时提示填写完整信息；

用户名不存在时提示“用户名不存在”；

密码错误时提示“密码不正确”；

用户名密码正确时登录成功，跳转评分页面。

2.注册功能

信息不完整时提示补充内容；

用户名重复时提示“用户名已存在”；

密码不一致时提示修改；

信息正确时成功注册新用户，跳转登录页面。

3.重置密码

信息不完整时提示补充内容；

用户名不存在时提示“用户名不存在”；

密码不一致时提示修改；

信息正确时成功重置密码，跳转登录页面。

经过测试，前后端功能配合良好，系统稳定，操作体验流畅。

# 第七章 总结与展望

随着互联网技术的飞速发展和人们生活水平的提高，人们对健康问题的关注日益增加。传统的健康信息传播渠道，内容太散、被传播对象不精准、用户查找不便捷。现有的健康服务网站对于实现用户所有的需求，仍具有明显的距离。实现一个社区向的健康信息网站是一个美好的愿景，其旨在人们快速地获取想要的健康知识、及时地收到疾控信息、方便地交流疾病经历，可以在社会中发挥不俗的积极作用，更好地服务于人们。

本文是关于一款“基于JavaWeb健康咨询网站”的设计与实现。本文的主要内容是介 绍本论文的研究背景和意义，通过对传统健康服务网站的现状与现实用户的需求进行分析；介绍了现在流行的技术，包括HTML&CSS，JavaScript，JSP 技术，B/S架构，MVC，Maven，Servlet，JDBC。借助excalidraw绘图工具和流程图，完成了初步的技术要求和预期发展目标；系统的设计从快速响应和人性化入手，详细地设计运行流程，并根据需求针对前端页面结构以及功能进行了详细设计，使整个项目分为不同的功能模块。通过测试证明了设计在功能、性能上达到了预期需求，稳定性高，可靠性强。

我在项目中主要负责的便是业务逻辑层和资源管理层的开发。通过多次在项目中完成数据库传值的操作，我对一个系统的运行逻辑有了深刻地认识。在开发期间，我通过在网络中寻找运用了各个技术的不同优秀项目，将他们的运用总结在一起，学习并使用在代码中，补充了自己的知识点，实现了更多的功能，使项目更加丰富的同时，也提高了自身的编程能力。

尽管本文已经成功设计并实现了基于JavaWeb的健康自查网站，但是仍有可进一步研究和改进的地方。例如，提高业务逻辑层与前端页面的交互密度，减少页面的跳转次数，增加动态内容的生成。其次，用户对网站内容的可更改权限也仍有进步空间，需要更加合理规划，以便长久运营。

# 参考文献

张波,张福炎.基于JSP技术的Web应用程序的开发[J].计算机应用研究,2001,(05):99-101.

唐海涛.MVC在多层Web体系结构中的应用研究[D].河海大学,2003.

李俊杰.Maven在企业Java软件产品中的应用[J].电脑知识与技术,2011,7(07):1562-1565+1588.

江日念,林霞,乔德新.Maven在Java项目中的引入及应用[J].电脑知识与技术,2013,9(21):4842-4847+4851.

杨新艳,于伟涛.基于Maven的轻量级Java软件开发研究[J].科技传播,2015,7(17):134-135.DOI:10.16607/j.cnki.1674-6708.2015.17.085.

任晓蕾,魏海平,林涛,等.基于Web数据库动态访问技术的Java实现[J].计算机与现代化,2000,(02):65-69+86.

樊胜.C/S与B/S的结构比较及Web数据库的访问方式[J].情报科学,2001,(04):443-445.

# 致谢

来到了毕业设计的最后阶段，这代表着机电生活迎来了尾声。我非常感谢这几年来所有的专业课老师，他们共同培养了我的编程能力，系统的所有逻辑代码都是源自他们传授我的宝贵知识。感谢指导老师对本项目的帮助，在项目停滞不前时总能为我们带来指点，也感谢指导老师为我们的论文进行修改。感谢我的组员们，如果失去了你们任意一位，这个项目将无法实现，感谢你们在我遇到困难时给我陪伴与鼓励，感谢你们与我一起协作完成毕业设计。