

成绩:

江西科技师范大学

课程设计（论文）

题目（中文）：基于 Web 客户端技术的个性化 UI 的设计和编程

（外文）：Customized UI design and Programming based on Web
client technology

院（系）：元宇宙产业学院

专 业：计算机科学与技术

学生姓名：颜廖文瑾

学 号：20213874

指导教师：李健宏

2024 年 4 月 27 日

目录

1. 引言	1
1.1 毕设任务分析	2
1.2 研学计划	3
1.3 研究方法	4
2. 技术总结和文献综述.....	5
2.1 Web 平台和客户端技术概述	5
2.2 项目的增量式迭代开发模式	6
3. 概述内容设计要点.....	8
3.1 分析和设计	9
3.2 项目的实现和编程	10
3.3 项目的运行和测试	12
3.4 项目的代码提交和版本管理	13
4. 应用响应式设计技术开发可适配窄屏和宽屏的 UI.....	14
4.1 分析和设计	14
4.2 项目的实现和编程	15
4.3 项目的运行和测试	17
4.4 项目的代码提交和版本管理	18
5. 移动互联时代的响应式设计和窄屏代码实现	19
5.1 分析和设计	19
5.2 项目的实现和编程	20
5.3 项目的运行和测试	22
5.4 项目的代码提交和版本管理	23
6. 适用移动互联时代的响应式设计	23
6.1 分析和设计	23
6.2 项目的实现和编程	24
6.3 项目的运行和测试	28
6.4 项目的代码提交和版本管理	28
7. 个性化 UI 设计中鼠标模型以及通用 UI 设计.....	29
7.1 分析和设计	29
7.2 项目的实现和编程	30
7.3 项目的运行和测试	31
7.4 项目的代码提交和版本管理	32
8. UI 的个性化键盘控制	32
8.1 分析和设计	32
8.2 项目的实现和编程	33
8.3 项目的运行和测试	35

8.4	项目的代码提交和版本管理	36
9.	本项目中的高质量代码.....	36
10.	用 gitBash 工具管理项目的代码仓库和 http 服务器.....	37
10.1	经典 Bash 工具介绍	37
10.2	通过 github 平台访问项目	38
	参考文献	39

WebAPPUI 开发技术

摘要：在过去十年间，HTML5 作为 Web 标准的核心，凭借其跨平台能力和开源特性，在软件开发领域获得了广泛的认可和应用。本研究项目依托于 Web 客户端技术，深入探讨并实践了程序设计和软件开发。通过深入研究技术文献和参与开发者社区的讨论，我们成功设计并实现了一个具有个性化用户界面的应用程序。在开发过程中，我们直接应用了 Web 客户端的基础 API，并利用 HTML、CSS 和 JavaScript 等技术分别进行内容构建、界面设计和交互功能实现。此外，项目还实施了响应式设计策略，以满足移动互联网时代多样化的设备需求。项目还采用了面向对象的编程方法，通过统一的代码库构建了一个指针模型，实现了对不同输入设备的兼容，从而提高了代码的质量和维护性。在项目管理层面，本项目采取了增量式开发方法，通过多轮迭代（包括分析、设计、实现和测试）来逐步优化和完善产品。在开源和代码共享方面，我们采用 Git 进行版本控制，经过多次代码重构和修正，最终将项目代码发布至 GitHub 平台。利用 GitHub 提供的 HTTP 服务，我们在全球范围内部署了应用程序，用户可以通过网络链接或扫描二维码轻松访问，实现了跨平台的便捷访问。

关键词：计算机科学与技术；web 前端开发；html；css；git；

1. 引言

设计和编程一个基于 Web 客户端技术的个性化用户界面（UI）是一个涉及多个学科的综合学习过程。首先，学习计划应该从基础的 HTML、CSS 和 JavaScript 开始，逐步过渡到更高级的前端框架如 React、Vue 或 Angular，同时学习 CSS 预处理器和前端构建工具。随着对前端技术的深入理解，可以开始实践项目，设计和实现个性化 UI 组件，并学习版本控制工具如 Git。进一步的学习包括后端技术、数据库技术以及 API 设计，这些都是实现个性化 UI 不可或缺的部分。高级阶段可以探索使用机器学习技术来增强个性化体验，并通过数据分析和用户行为分析来优化设计。

技术路线方面，项目应该从需求分析开始，明确项目目标和用户需求，然后进行技术选型，选择适合项目需求的前端框架和后端技术。设计阶段需要创建原型和设计稿，之后是前端和后端的开发工作，包括实现数据处理的业务逻辑。集成测试确保前端和后端的无缝集成，性能优化则关注加载时间和响应速度。项目完成后，需要进行部署上线，并通过用户测试和反馈收集来进行迭代改进。

参考资料的选择也非常关键，可以从基础的 Web 开发书籍开始，如"HTML & CSS: Design and Build Websites"和"JavaScript: The Good Parts"，同时利用在线资

源如 MDN Web Docs、W3Schools 和 FreeCodeCamp 进行自学。框架的官方文档也是不可或缺的学习资源。设计工具如 Adobe XD、Sketch 和 Figma 可以帮助设计 UI/UX。

研究方法上，可以通过文献综述来了解现有的个性化 UI 设计和实现方法，案例研究可以提供实际应用的视角。用户调研是理解用户需求的关键，原型测试可以收集用户反馈，A/B 测试有助于选择最佳设计方案。数据分析工具可以帮助分析用户行为和偏好，而机器学习则为个性化体验提供了新的可能。

通过这样的学习计划、技术路线、参考资料和研究方法，可以系统地掌握个性化 UI 设计和编程所需的知识和技能，最终实现一个既美观又实用的 Web 客户端应用。

1.1 毕业设计任务分析

通用的毕业论文（设计）管理手册中，表述为“毕业论文（设计）”，这在字面上似乎更侧重于论文，而对毕业设计作品的要求则没有明确强调。然而，对于软件开发专业的毕业设计，这种表述可能不够精确，因为通常需要先完成软件的开发，然后才能撰写出相应的论文。

在计算机专业软件开发的毕业论文中，可以采用截图与文本相结合的方式，展示软件的功能和效果，以证明其真实性。此外，论文中还可以直接提供访问毕设作品的网址或二维码，使读者能够直接体验和使用作品。这种方法不仅增强了论文的可信度，也体现了计算机专业论文的特色，这是仅依靠文本描述所难以达到的。

计算机科学技术专业的本科生，在学业即将结束之际，设计并开发一个与专业相关的作品是十分必要的。这不仅是对所学知识的回顾和总结，也是对课程体系核心内容的梳理，能够展示学生的真实能力。

在毕业设计中，应涵盖面向对象程序设计语言、数据结构与算法、操作系统、软件工程等核心课程的理论。这些理论在以往的学习中可能显得抽象，主要因为它们侧重于理论知识。通过与导师的沟通，一致认为将这些核心课程的关键知识应用到实践中，可以在理解和技术两个层面提升专业性。

毕业设计的内涵在于将大学期间所学的理论知识在实践层面进行一次综合

的演练和总结。同时，也需要学习并应用当前流行的技术，形成对计算机软硬件体系的系统性、专业性理解，并在此基础上撰写毕业论文，这正是毕业论文的内涵所在。

深刻理解计算机系统对于计算机专业开发者至关重要，这不仅是即将成为国家现代化建设工程师的特色，也是与其他专业不同的方面。从其他专业的角度来看，计算机专业的学生对计算机系统的理解不应停留在表面，而应更深入地接近计算机的本质，对技术的掌握也应尽可能接近其底层和基本原理。

1.2 研学计划

毕设的完成分为两个阶段。首阶段涉及选择一条技术实践路线，整合并学习核心技术，参考导师的案例项目，理解技术间的相互关系及其在项目中的角色和分工。重点在于通过项目实施提升编写高质量代码的能力。

技术基本实现后，可视为实践与理论的结合，进入第二阶段，即开发毕设软件。这一阶段遵循软件工程标准，包括：定义和分析问题、设计技术解决方案、根据方案设计流程和编写代码、实现技术部署、调试代码、测试软件和性能调优。步骤 3 和 4 可能发现前面步骤的问题，导致在设计、编码和测试阶段的多次迭代，以修正设计失误或代码错误。大部分工作集中在构建代码体系和落实软件架构上。

本科毕设与个人开发相似，缺乏经验丰富的团队支持，设计细化优化与代码部署经常交替进行。细化优化是工程师在微观层面的工作，而部署则是设计师确保宏观设计符合需求的任务。

开发过程中产生的大量文档，经过总结并与专业理论结合，可形成毕设论文。这一路径允许学生通过实践加深对理论的理解，从而深化本科学习的理论体会，实现从实践到理论再到最佳实践的升华。

技术路线是指为实现复杂工程而制定的计划或规划，强调专业性和逻辑性。它通常使用专业术语表达，并展示事物的因果或层次关系，常用于证明工程项目或研究的可行性。

1.3 研究方法

在本人的毕业设计中，试图解决的问题也被定义为不同层级和类型的模型。模型研究法相对抽象，与具体的计算机语言无关。在毕设中，本人尝试采用统一建模语言（UML）来构建抽象模型。本人认为，使用 UML 模型研究法与面向对象的程序设计方法在目标上是一致的，只是分析和表述问题的角度不同。

因此，在毕设中，本人采用了 UML 对问题进行建模。尽管 UML 较为抽象，设计起来有一定难度，但使用 OOP 程序设计来实现这些模型则更为具体直观。通过熟悉的 OOP 语言和代码运行环境来运行和调试模型，可以发现模型设计的问题和缺陷。

对于计算机科学专业的本科生而言，可能先通过编程来建立和研究模型，代码运行成功后，再利用 UML 来绘制模型，作为代码的文档可能更为合理。毕竟，本科生的思维尚处于成熟阶段，无法与高级程序员或系统架构师的抽象能力和丰富经验相媲美。在学术研究中，文献法是一种基础的学习方法，其核心在于通过阅读和分析前人的著作来深入理解并掌握特定领域的知识与技术。对于 Web 应用开发领域的学生而言，文献法是不可或缺的，无论在编程技能的哪个成长阶段。知识的载体形式多样，包括但不限于书籍、在线文档、社区论坛、期刊和会议报告等。

在毕业设计中，资料来源可分为几类：首先是以"O'Reilly Media"和"Manning Publications"为代表的出版集团，它们出版了众多计算机科学领域的经典著作。这些书籍在全球范围内被广泛传播，尤其是在互联网上的开发者社区中。尽管这些书籍通常为英文原版，但在知识共享的时代背景下，其作者应能理解并接受发展中国家学生免费获取知识的需求。

其次是在线文档和社区论坛，如 W3C 和 MDN，它们分别代表了科学和工程两个层面的知识积累，提供了大量关于 Web 技术发展的公开文档。W3C 主要关注 Web 国际标准和最佳实践，而 MDN 则更侧重于 Web 应用的教学和讨论。

再者是知网平台，它为学生提供了论文写作规范和理论指导，尽管对于软件开发能力的直接影响有限，但对于毕业论文的撰写具有重要参考价值。

对于学习编程的学生而言，模型研究法是另一种重要的研究方法。这种方法涉及对软硬件对象的深入分析，将它们抽象为模型，并通过程序化和数据化的过

程转化为代码。在面向对象编程（OOP）的分析和开发过程中，问题被定义为不同层级的模型。模型研究法与 OOP 的目标一致，只是分析和表述问题的角度不同。在毕业设计中，UML（统一建模语言）被用来建立抽象模型，尽管 UML 模型的设计具有一定的难度，但通过 OOP 语言对模型进行程序设计则更为直观。通过运行和调试代码，可以发现并修正模型设计中的问题和缺陷。对于本科生而言，建议在编写代码并建立模型后，再使用 UML 作为文档资料进行模型的绘制，这更符合本科生的思维发展阶段。

2. 技术总结和文献综述

网络之父蒂姆·伯纳斯-李在创造网络的基本技术架构之后，成立了万维网联盟（W3C）^[1]。该组织在 2010 年之后推出了 HTML5 的国际标准，并且与欧洲的 ECMA 组织维护的 ECMAScript 国际标准相结合，几乎完美地实现了全球开发者统一开发平台的理想。至今，科学家和网络行业依然在努力完善这一伟大而崇高的目标。我的毕业设计项目应用的技术路线是学习网络标准和网络技术，学习编写网络程序和使用相关工具，最终构建一套高质量的跨平台运行的应用程序代码。

2.1 Web 平台和客户端技术概述

1989 年，蒂姆·伯纳斯-李爵士发明了万维网（参见原始提案）。他创造了“万维网”这个术语，编写了第一个万维网服务器“httpd”，以及第一个客户端程序（浏览器和编辑器），名为“World Wide Web”，在 1990 年 10 月。

他编写了“超文本标记语言”（HTML）的第一个版本，这是一种文档格式化语言，具有超文本链接的能力，成为了网络的主要发布格式。随着网络技术的传播，他最初对 URI、HTTP 和 HTML 的规范被进一步细化，并在更广泛的圈子中进行了讨论。

1994 年，在众多公司不断增加对网络的投资的推动下，决定成立万维网联盟。蒂姆·伯纳斯-李爵士开始领导万维网联盟团队进行关键工作，以促进一个一致的架构，适应网络标准快速发展的步伐，用于构建网站、浏览器和设备，以

体验网络所提供的一切。

在创立万维网联盟时，蒂姆·伯纳斯-李爵士创建了一个同行社区。网络技术的发展速度非常快，因此组建一个单一的组织来协调网络标准至关重要。蒂姆接受了拥有联盟经验的麻省理工学院的邀请，来托管 W3C。他从一开始就要求 W3C 具有全球影响力。

万维网（Web）是 World Wide Web 的缩写。大多数人用“Web”代替“World Wide Web”，我们也遵循这一惯例。Web 是由文档组成的集合，这些文档被称为网页，并在全球范围内主要由计算机用户共享。不同类型的网页有不同的功能，但它们都至少会在计算机屏幕上显示内容。这里的“内容”指的是文本、图片以及用户输入机制，如文本框和按钮^[2]。

Web 编程是一个广泛的领域，不同类型的 Web 编程由不同的工具实现。所有工具都与核心语言 HTML 一起工作，因此几乎所有的 Web 编程书籍都在一定程度上描述了 HTML。这本教材深入讲解了 HTML5、CSS 和 JavaScript。这三种技术被公认为是客户端 Web 编程的基石。在客户端 Web 编程中，所有网页计算都在最终用户的计算机（客户端计算机）上执行。Web 应用的程序设计体系主要由 HTML、CSS 和 JavaScript 三大语言构成，这三者的结合体现了人类社会分工协作的智慧。它们各自独立而又相互关联，共同实现了一个信息系统的全面描述和控制。具体而言，HTML 负责构建页面的基本结构，CSS 负责塑造页面的外观呈现，而 JavaScript 则赋予页面动态的行为响应^[3]。这一分工方式可以被视作对信息系统进行细致化、专业化的管理。若以 MVC（Model-View-Controller）设计模式来类比，HTML 可视为 Model，即数据模型的基础构建；CSS 作为 View，负责界面的视觉表现；而 JavaScript 则扮演 Controller 的角色，结合前两者，在微观和功能层面实现对代码的控制和交互^[4]。

2.2 项目的增量式迭代开发模式

增量式迭代开发模式是一种将软件开发过程划分为多个小规模、可管理的阶段，每个阶段都产生一个可执行的软件版本的方法。这种模式通过逐步构建和完善产品，允许开发团队在每个迭代周期结束时，获得一个功能更加丰富的软件版本。选择增量式迭代开发模式的原因在于其能够显著降低项目风险，通过早期发

现和解决问题，减少后期大规模修改的需求。此外，该模式强调持续的用户反馈，使开发方向能够及时调整以满足用户需求，同时增强了开发过程的灵活性和市场适应性。增量式迭代开发模式还支持快速响应市场变化，允许产品快速推向市场，获得竞争优势。在每个迭代中进行的测试有助于持续提高软件质量，而客户在整个开发过程中的参与则确保最终产品符合客户的期望和业务目标。资源优化和知识积累也是选择增量式迭代开发模式的重要原因，它们确保了关键任务的优先处理和团队开发效率的提升。总体而言，增量式迭代开发模式因其在风险管理、质量保证、客户参与和市场适应性方面的优势，成为许多软件开发项目的首选方法。因此，在这次项目的开发中也使用了增量开发的模式。

增量模型采用逐步构建的方式开发软件。开发团队首先创建一个基础版本，该版本虽简单但包含了整个系统的核心框架。如图 1.1 所示，增量模型的核心思想是逐步添加细节和功能。在每个迭代中，新的功能被集成并进行测试。如果发现问题，开发团队可以迅速定位到最新添加的部分。这种模型的优势在于问题易于追踪和修复，且可以逐步完善系统，直到所有功能都实现^[5]。

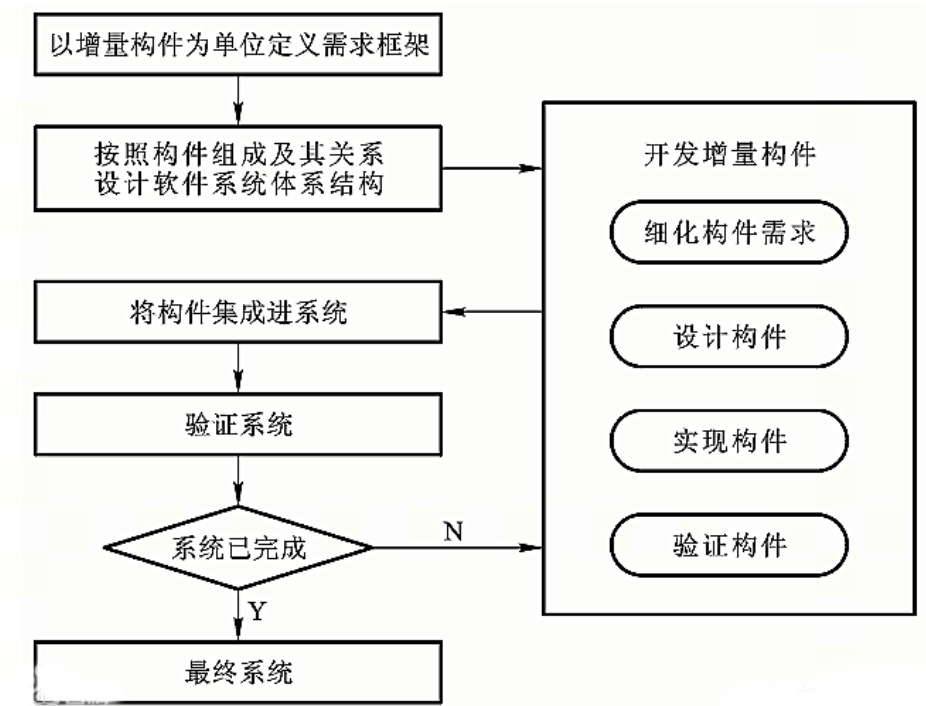


图 1.1 增量模型示意图

3. 概述内容设计要点

随着信息技术的飞速发展,Web 应用已成为人们日常生活中不可或缺的一部分。为了满足用户对个性化和交互性日益增长的需求,本项目致力于开发一个具有高度互动性和个性化用户界面的 Web 应用程序。研究背景基于对现有 Web 应用的分析,发现用户对于界面的美观性、易用性和响应速度有着更高的期待。因此,本项目的目标是设计并实现一个既美观又实用的 Web 应用,以提升用户体验并满足市场的需求。

在项目实施过程中,我们首先进行了市场调研和用户需求分析,以确保开发的方向与用户的实际需求相匹配。随后,我们规划了详细的项目时间表和里程碑,确保项目按计划推进。在工作量方面,我们投入了大量的时间和资源进行前端技术的研究与学习,包括但不限于 HTML、CSS 和 JavaScript 的深入掌握。

在技术实现方面,我采用了现代 Web 开发的最佳实践。通过 HTML 语言,我们构建了内容的结构框架,确保了信息的组织和呈现方式符合逻辑和语义化原则。CSS 语言的应用使我们能够设计出既美观又符合品牌风格的用户界面,提升了应用的视觉吸引力。JavaScript 语言的编程则赋予了应用丰富的交互功能,使用户的操作更加直观和流畅。此外,我还采用了响应式设计原则,确保应用在不同设备和屏幕尺寸上都能提供一致的用户体验。

综合这些方法,我的工作不仅提升了 Web 应用的技术标准,也推动了用户界面设计的创新。通过这些技术的融合应用,我们为用户提供了一个既美观又实用的 Web 平台,这在当前竞争激烈的市场环境中具有重要的意义。我的工作不仅满足了用户对高质量 Web 应用的需求,也为 Web 开发领域提供了新的思路和解决方案。整个项目的蓝图截图如图 2.1 所示。



图 2.1 项目蓝图示意图

3.1 分析和设计

本项目在内容设计上遵循了人们习惯的“三段论”模式，旨在通过清晰的结构和层次来呈现信息，同时确保用户体验的流畅与舒适。首先，项目以一个醒目的标题性信息作为起始点，这可以是精心设计的 logo 或是简洁明了的文字标题。这一步的目的在于迅速吸引用户的注意力，引导他们进一步探索。

接下来，展示的主要内容区域是整个设计的核心。这里，“内容为王”是我们坚守的理念，意味着无论是文字、图片还是其他多媒体元素，都将以高质量、高价值的内容为主导。这一区域不仅承载着项目的核心信息，更是与用户进行深度互动的关键。我们精心策划和制作内容，确保每一部分都能与用户产生共鸣，提供有价值的信息和体验。

最后，在内容的尾部，我们设置了附加信息区域。这个区域用于展示一些用户可能关心的细节变化，如更新日志、版权声明等。虽然这些信息不是主要焦点，但它们对于提升用户体验和信任度同样重要。通过这一区域的设置，我们旨在为用户提供更加全面、完善的信息服务。

如图 3.1 所示，本项目的内容设计清晰明了，遵循了“三段论”的模式，确保了信息的有效传达和用户体验的优化。这种设计方式不仅符合人们的阅读习惯和认知规律，更能够在众多项目中脱颖而出，成为用户喜爱的优秀作品。

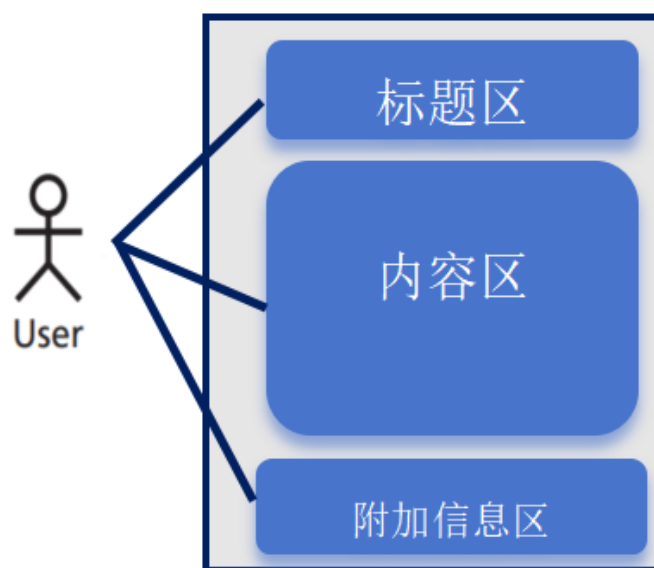


图 3.1 项目用例图

网页设计者通常制定一套通用的布局原则，而不是为每种显示设备定制网页版本。他们允许浏览器基于当前的硬件环境来决定页面的具体显示方式。这意味着网页设计不会涵盖所有细节，例如，设计者可以定义段落的结构，但不会具体规定每行文字的精确长度或段落首行是否缩进。

这种显示细节的灵活性导致了一个现象：同一网页在不同的浏览器或不同硬件配置的计算机上可能会呈现出不同的视觉效果。例如，如果两个屏幕的宽度不同，那么显示的文本行长度或图像尺寸可能会有所差异。关键在于，网页提供了一个关于期望展示效果的大致框架；而具体的显示细节则由浏览器在渲染页面时决定。因此，即使是同一个网页，在不同的设备或浏览器上也可能呈现出细微的差别。

3.2 项目的实现和编程

首先使用 **HTML** 代码呈现一个典型的网页结构，其中包含了头部、导航、主体内容和页脚四个主要部分。在<header>标签内，通过<p>元素展示了“剑桥雅思系列课程@masterLijh”这一标题性信息，既表明了页面的主题，也暗示了内容的发布者。紧接其后的<nav>元素虽然当前仅包含一段提示性文字“海量经典正向您传送过来，请耐心等待...”，但预示着此处将作为未来导航链接的容器，为用户提供网站不同部分的跳转路径。<main>元素作为页面核心内容的承载区，目

前为空，预示着后续将在此处填充具体的文章、图片或其他多媒体内容。最后，在<footer>标签内，通过<p>元素展示了版权信息，包括作者“颜廖文瑾”、所属机构“江西科技师范大学”以及版权期限“2022--2025”，这不仅体现了对知识产权的尊重，也为网页增添了规范性和权威性。这段代码虽然简洁，但已搭建起了一个完整的网页框架，为后续内容的填充和功能的扩展奠定了基础。

```
<body>
  <header>
    <p id="book">
      剑桥雅思系列课程@masterLijh
    </p>
  </header>
  <nav>
    海量经典正向您传送过来，请耐心等待...
  </nav>

  <main>

</main>

  <footer>
    <p id="statusInfo">
      Copyright from 颜廖文瑾 江西科技师范大学 2022--2025
    </p>
  </footer>
</body>
```

在第一次迭代中，我同时完成了一个 CSS(层叠样式表)片段用于定义 HTML 文档中特定元素的样式。

此 CSS 代码定义了一个全局样式规则，针对 body 元素。它设置了整个文档的主体文本大小为 26 像素 (font-size: 26px;)。这意味着页面上的所有文本，除非被其他更具体的样式规则覆盖，都将采用这个字体大小。

同时代码定义了三个独立的样式规则，分别针对 header、footer 和 nav 元素。这三个元素在 HTML 文档中通常用于表示页面的头部、页脚和导航栏。

对于 header、footer 和 nav 元素，每个规则都设置了相同的两个样式属性。首先是 border 属性，它定义了一个 2 像素宽、实线样式、蓝色的边框 (border: 2px

`solid blue;`)。这将使得这三个元素在页面上呈现出一个明显的蓝色边框，有助于区分它们与其他内容。

第二个属性是 `height`，它指定了这三个元素的高度为 200 像素 (`height: 200px;`)。这意味着无论这三个元素内部的内容有多少，它们在页面上都将占据固定的 200 像素高度空间。如果内容过多或过少，它们可能会超出或低于这个高度（取决于内容是否溢出或是否使用了适当的垂直对齐方式），但元素本身的高度限制将是 200 像素。

```
<style>
  body {
    font-size: 26px;
  }

  header {
    border: 2px solid blue;
    height: 200px;
  }

  footer {
    border: 2px solid blue;
    height: 200px;
  }

  nav {
    border: 2px solid blue;
    height: 200px;
  }
</style>
```

通过 CSS 为 `body`、`header`、`footer` 和 `nav` 元素提供了具体的样式规则，包括字体大小和边框及高度的设置。这些规则将直接影响这些元素在项目的页面上的外观和布局。

3.3 项目的运行和测试

在测试与验证阶段，我将这段代码上传到 `github` 上并通过浏览器生成了该网页的二维码，如图 4.1 所示。在手机端上打开该网页的效果如图 4.2 所示。在电脑端上的效果图如图 4.3 所示。

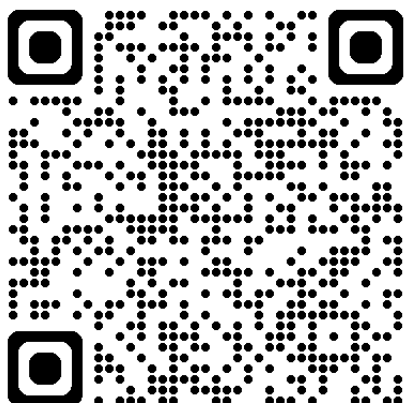


图 4.1 第一次迭代代码网页二维码



图 4.2 第一次迭代代码手机端效果图

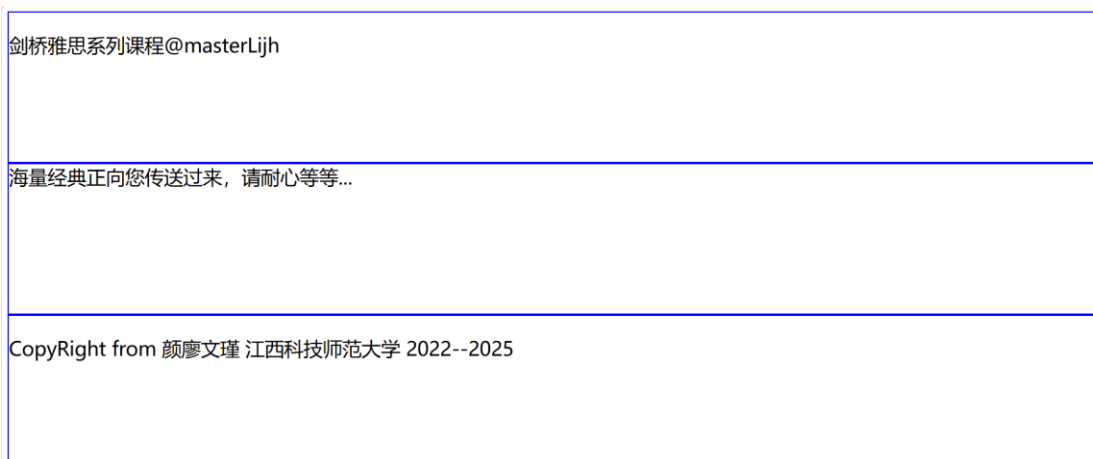


图 4.3 第一次迭代代码 PC 端效果图

3.4 项目的代码提交和版本管理

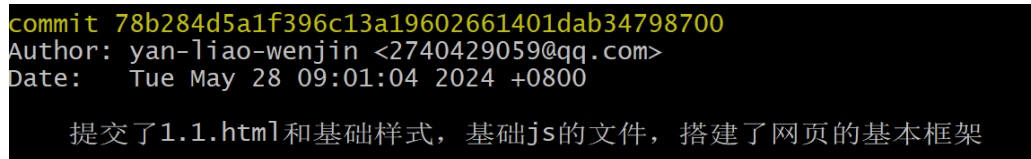
本项目的文件管理采用 gitBash 工具，旨在首次迭代中确立项目的基础文件架构，并完成代码仓库的初步设置，包括录入开发者的姓名与电子邮件地址等基本信息。在启动 gitBash 的命令行界面后，需依次执行以下指令序列：

```
$ cd /  
$ mkdir webUI  
$ cd webUI  
$ git init  
$ git config user.name yanliaowenjin  
$ git config user.email 2210292957@qq.com  
$ touch index.html myCss.css
```


完成 index.html 和 myCss.css 文件的编码工作，并在测试确认无误后，进行以下步骤以提交代码：

```
$ git add index.html myCss.css  
$ git commit -m 提交了 1.1.html 和基础样式...
```

提交后的日志如图 4.4 所示。



```
commit 78b284d5a1f396c13a19602661401dab34798700  
Author: yan-liao-wenjin <2740429059@qq.com>  
Date: Tue May 28 09:01:04 2024 +0800  
  
提交了1.1.html和基础样式，基础js的文件，搭建了网页的基本框架
```

图 4.4 第一次迭代代码提交效果图

4. 应用响应式设计技术开发可适配窄屏和宽屏的 UI

4.1 分析和设计

在移动互联时代，用户终端的多样性是其核心特征之一。随着智能手机、平板电脑、智能手表等设备的广泛使用，用户接入互联网的方式变得多样化。这种多样化对网页设计提出了新的要求，即网页需要能够适应不同尺寸和分辨率的屏幕。响应式设计应运而生，它通过 CSS 语言和 JavaScript 语言实现网页的自适应性，以满足不同设备的需求。

响应式设计中的流体布局技术使用百分比而非固定像素来定义元素宽度，使得布局能够根据屏幕尺寸的变化而自动伸缩。同时，灵活图片和媒体技术通过设置图片和视频元素的最大宽度为 100%，确保它们能够适应其容器的宽度，避免超出屏幕边界。CSS3 的媒体查询功能允许开发者根据设备的特性应用不同的样式规则，实现设计对特定设备或屏幕大小的优化。此外，可伸缩的矢量图形（SVG）技术使用 XML 描述矢量图形，能够在不同分辨率下保持图形质量，适合多种显示需求。

JavaScript 和 jQuery 的使用可以进一步增强响应式设计，通过检测设备的屏幕尺寸和方向，动态调整布局或执行特定功能。此外，存在多种响应式框架和库，如 Bootstrap、Foundation 等，它们提供预定义的响应式组件和工具，简化了开发过程。最后，通过在多种设备和模拟器上进行测试，确保网页在不同环境下都能提供良好的用户体验，是实现响应式设计不可或缺的一环。通过这些技术的综合

应用，可以构建出能够自适应不同用户终端的网页，满足移动互联时代对网页可访问性和视觉一致性的需求。

对比第一次迭代页面中增加了三个按钮，分别是导航 1，导航 2 和导航 3。并将背景图片置于整个页面上。第二次迭代的用例图如图 4.5 所示。

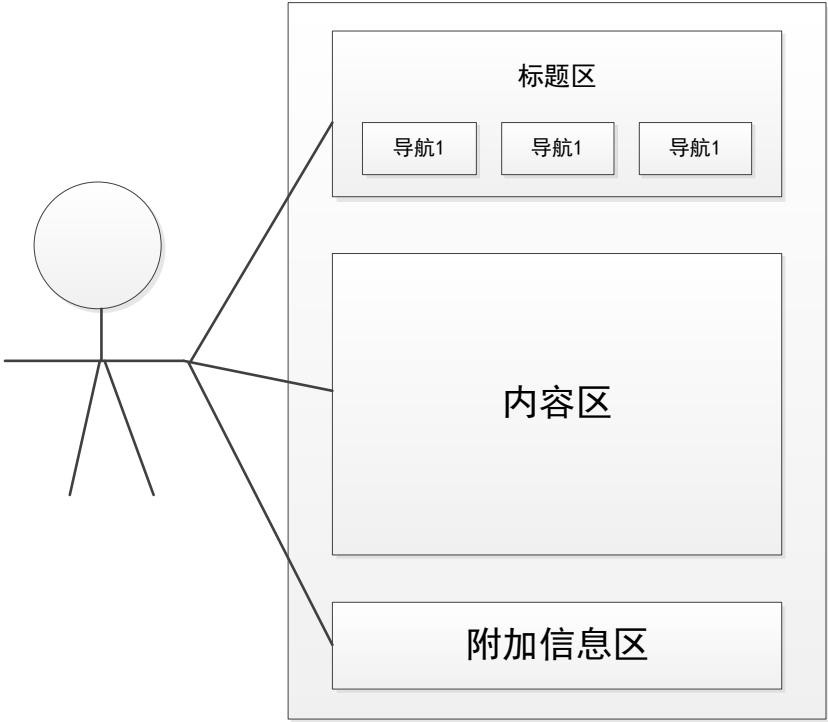


图 4.5 第二次迭代项目用例图

4.2 项目的实现和编程

在第三次迭代中，构成了一个基本的 HTML5 网页，它包括了文档类型声明、HTML 根元素、头部、主体和脚本。文档类型声明<!doctype html>告诉浏览器这是一个 HTML5 文档。HTML 根元素<html lang="zh">设置了页面的主要语言为中文。头部<head>中包含了字符集声明<meta charset="UTF-8">，确保页面可以正确显示各种字符；视口设置<meta name="viewport">确保页面在不同设备上能够正确缩放；页面标题<title>定义了页面的标题，显示在浏览器标签上；内联样式<style>定义了页面的 CSS 样式，包括页面元素的布局、颜色、字体大小等。

```
<style>
  body {

    text-align: center;
  }
```

```

header {
  border: 2px solid blue;
  height: 10%;
  background-color: blue;
  color: white;
}
nav {
  border: 2px solid blue;
  height: 5%;
}
main {
  height: 75%;
  background-image: url(../lesson/CS.jpg);
  background-repeat: no-repeat;
  background-size: cover;
  background-position: center;
}
footer {
  border: 2px solid blue;
  height: 10%;
  background-color: black;
  color: white;
}

p#book {
  font-size: 1.8em;
  letter-spacing: 0.2em;
}

button {
  font-size: 1.2em;
  margin-right: 1em;
}

```

</style>

在页面的主体<body>中，<header>元素包含了一个带有特定 id 的段落<p id="book">，显示了页面的主题。紧接着的<nav>元素包含三个按钮，可能用于页面导航。<main>元素作为主要内容区域，背景设置了一张图片，并通过 CSS 属性确保图片覆盖整个区域且居中显示。<footer>元素包含了版权信息，显示了作者和版权年份。

最后，<script>标签内包含了 JavaScript 脚本，定义了一个 UI 对象，用于存储设备的宽度和高度，并根据这些尺寸动态调整页面的字体大小和整体布局，使得页面能够适应不同设备的显示需求。整体上，这段代码展示了一个具有响应式设计基础的网页，通过内联 CSS 和 JavaScript 实现了页面的基本样式和动态调整功能。

```
<script>
  var UI = {};
  UI.deviceWidth = window.innerWidth;
  UI.deviceHeight = window.innerHeight;
  document.body.style.height = UI.deviceHeight + 'px';
  document.body.style.fontSize = UI.deviceWidth/25 + 'px';
</script>
```

4.3 项目的运行和测试

这段代码创建了一个具有响应式设计特性的网页，通过 HTML 定义了页面的基本结构，包括头部、导航、主要内容区域和页脚。页面的视口通过`<meta>`标签进行了设置，确保了在不同设备上能够正确缩放，实现响应式布局。CSS 样式直接内联在`<style>`标签中，定义了页面元素的尺寸、颜色、背景图像和字体样式，其中使用了百分比来创建流体布局，使得页面元素可以根据屏幕尺寸自动调整大小。背景图像通过 CSS 的`background-size: cover;`属性实现覆盖整个`main`元素，同时保持图像不失真。JavaScript 脚本进一步增强了页面的响应性，根据设备的宽度动态调整字体大小，提升了不同设备上的可读性。页面的头部和页脚采用了统一的设计风格，通过颜色和边框样式强化了页面的视觉主题。此外，页面还包含了导航按钮，尽管具体的导航功能没有在代码中实现。页脚部分展示了版权信息，符合网页版权声明的标准做法。整体上，内联样式简化了样式管理，但可能会影响样式的复用和维护。这段代码综合运用了 HTML、CSS 和 JavaScript 技术，展示了一个简洁且具有基本响应式功能的网页设计。将项目代码上传至 github 后访问相应的链接，其二维码与实现效果如 4.6 和 4.7 所示。

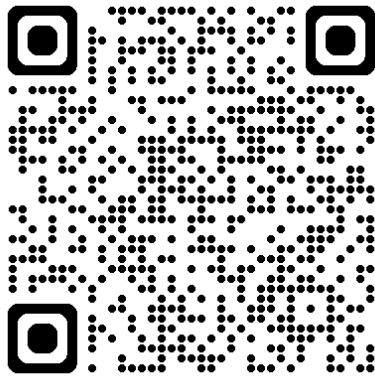


图 4.6 第二次迭代二维码



图 4.7 第二次迭代效果图

4.4项目的代码提交和版本管理

页面字体大小会根据屏幕宽度自动调整，以增强在各类设备上的阅读体验。统一的配色方案和边框设计贯穿于页面的顶部和底部，营造了一致的视觉风格。尽管代码中未具体展示，页面还包含了导航用的按钮。页脚部分的版权信息标注了作者姓名和版权有效期，遵循了网页版权声明的标准做法。CSS 样式直接嵌入在 HTML 中，虽然方便了样式的直接应用，但这种方法可能会限制样式在未来的重用和更新。综合来看，这段代码通过基础的 HTML、CSS 和 JavaScript 技术，实现了一个简洁且能够自适应不同屏幕尺寸的网页设计。与上一节步骤相同，在本地提交代码后的日志如图 4.8 所示。

```
commit 1f1a4bea7bfb00f0588ce1886501262d5db8ca8
Author: yan-liao-wenjin <2740429059@qq.com>
Date: Tue May 28 09:05:30 2024 +0800

与上一阶段比较，本阶段初次引入了em和%，这是CSS语言中比较高阶的语法，可以有效地实现我们的响应式设计，本阶段首次使用了JavaScript，首先创建了一个UI对象，然后把系统的宽度和高度记录在UI对象中，又计算了默认字体的大小，最后再利用动态CSS，实现了软件界面的全屏设置
```

图 4.8 第二次迭代提交日志

5. 移动互联时代的响应式设计和窄屏代码实现

计算机科学与技术专业的重要性和发展前景不容忽视。作为现代科技的基石，计算机科学与技术专业在推动科技创新、促进经济发展、改善社会生活等方面发挥着至关重要的作用。计算机所使用的显示硬件差异极大，显示器的尺寸和分辨率取决于成本。设计师们并未为每种类型的显示器都设计网页版本，而是选择为网页制定总体布局规范，让浏览器决定如何在特定计算机上显示页面。因此，网页不会提供太多细节。例如，网页的作者可以指定一组句子构成一个段落，但无法指定诸如行长的确切长度或是否缩进段落开头等细节。

允许浏览器选择显示细节会产生一个有趣的结果：当通过两个浏览器或两台具有不同硬件的计算机查看时，同一网页的外观可能有所不同。如果一个屏幕比另一个屏幕更宽，可显示的文本行长或图片大小也会有所不同。关键是：网页提供了关于所需呈现方式的总体指导；浏览器在显示页面时选择细节。因此，当在不同计算机或不同浏览器上显示时，同一网页的外观可能会略有不同。

5.1 分析和设计

相较于上一次，我通过 CSS 修改了网页的样式。* 选择器用于选择页面上的所有元素，并为它们设置了默认的外边距（margin）和文本对齐方式（text-align）。这种全局设置通常用于确保页面元素之间有适当的间距，并且文本内容居中显示。

针对页面中的特定元素，如 header、nav、main、footer，代码定义了各自的边框（border）、高度（height）和字体大小（font-size）。这些元素都设置了 3 像素的绿色实线边框，而它们的高度则根据页面布局的需要进行了分配。header 和 nav 各占页面高度的 10%，main 部分占据了 70%，而 footer 也占 10%。这种划分体现了典型的网页结构布局。

在 main 部分内部，代码使用了一个具有绝对定位（position: absolute）的 #box 元素。这个元素将始终相对于最近的已定位祖先元素（如果没有，则相对于初始包含块）进行定位，并且其右侧始终与包含块的右侧对齐。这种定位方式允许开发者在 main 区域内精确地放置内容。对于这次代码的迭代，如图 5.1

所示。

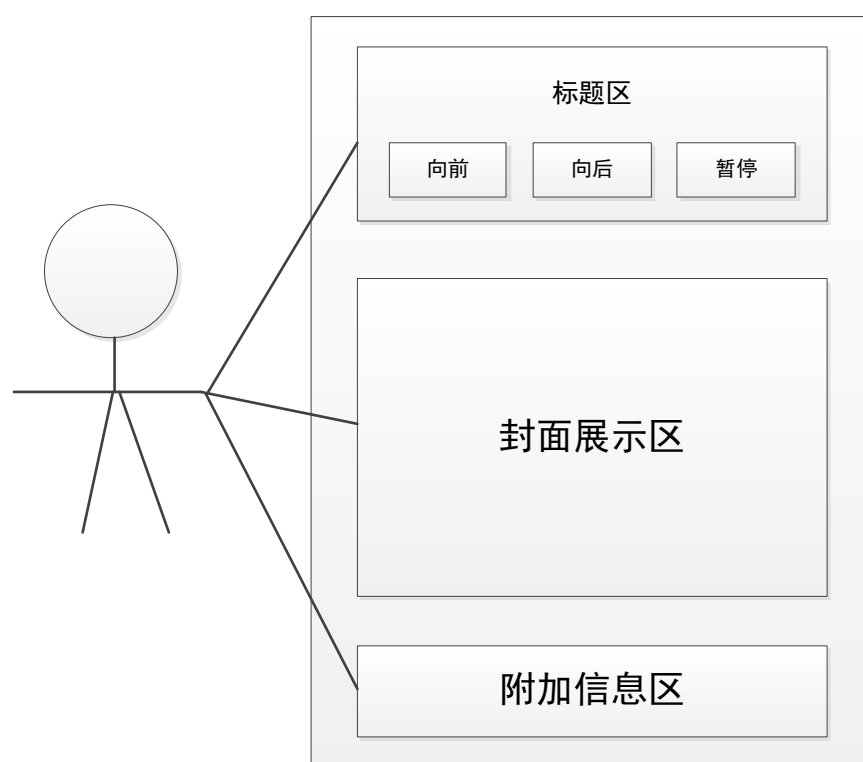


图 5.1 第三次迭代用例图

5.2 项目的实现和编程

`body` 元素被设置为相对定位 (`position: relative`)，这意味着它可以作为绝对定位元素的参考点。这在 `#aid` 元素的定位中得到了体现，该元素同样使用绝对定位，并且其位置是相对于 `body` 元素来确定的。其中，`#bookface` 元素被赋予了 80% 的宽度和高度，带有 1 像素的红色实线边框，以及浅桃色的背景色 (`blanchedalmond`)。 `margin: auto;` 的设置使得这个元素在其包含块内水平居中。

通过 CSS 来定义网页的基本结构和样式，包括元素的大小、位置、边框、字体等属性。这种结构化的样式定义有助于创建出既美观又易于维护的网页。具体代码如下所示：

```
<style>
*{
    margin: 10px;
    text-align: center;
}
header{
    border: 3px solid green;
```

```

    height: 10%;
    font-size: 1em;
}
nav{
    border: 3px solid green;
    height: 10%;
}
main{
    border: 3px solid green;
    height: 70%;
    font-size: 0.8em;
    position: relative;
}

#box{
    position: absolute;
    right: 0;
    width: 100px;
}

footer{
    border: 3px solid green;
    height: 10%;
    font-size: 0.7em;
}
body{
    position: relative;
}
#aid{
    position: absolute;
    border: 3px solid blue;
    top: 0px;
    left: 600px;
}
#bookface{
    width: 80%;
    height: 80%;
    border: 1px solid red;
    background-color: blanchedalmond;
    margin: auto;
}
</style>

```

与第一次比较, 我首次使用了 JavaScript, 这段代码定义了一个自定义的 \$ 函

数，该函数旨在简化对 DOM 元素的选择过程，同时设置了两个事件监听器以捕获用户在页面上特定元素（具有标识符 bookface）上的鼠标行为。代码初始化了几个与鼠标交互相关的变量，包括 `mouse.isDown`、`mouse.x` 和 `mouse.deltaX`。

接下来我为 `bookface` 元素添加了 `mousedown` 和 `mousemove` 事件监听器。当用户在 `bookface` 元素上按下鼠标时，`mousedown` 事件监听器会被触发，并捕获鼠标在页面上的当前坐标，随后这些坐标信息会被输出到控制台，并替换 `bookface` 元素的文本内容。类似地，当用户在 `bookface` 元素上移动鼠标时，`mousemove` 事件监听器会捕获鼠标的坐标并更新相应的输出。

我还定义了一个自定义的 `$` 函数，该函数接受一个字符串参数，用于通过 ID 或 CSS 选择器选择 DOM 元素。如果参数不是字符串类型，函数将抛出一个错误。函数首先尝试通过 ID 选择元素，如果未找到，则使用 CSS 选择器进行查找。如果两种方式都无法找到对应的元素，函数将再次抛出一个错误。这种自定义的 `$` 函数增强了代码的复用性和可读性。

5.3 项目的运行和测试

我通过这段代码通过自定义函数和事件监听器实现了对用户鼠标行为的捕获和响应，为后续的交互设计提供了基础。该页面的二维码和在手机端实际效果如图 5.2 和图 5.3 所示。在 PC 端的效果如图 5.4 所示。

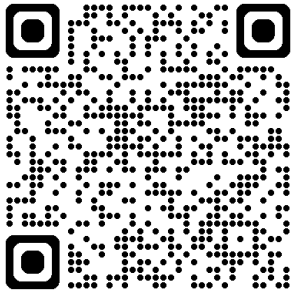


图 5.2 第三次迭代代码网页二维码

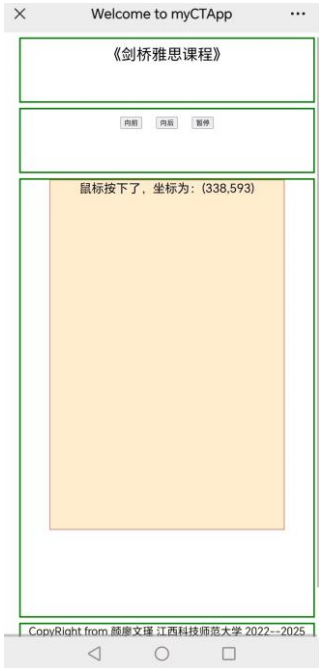


图 5.3 第三次迭代代码网页效果



图 5.4 第三次迭代代码 PC 网页效果

5.4 项目的代码提交和版本管理

在本地提交代码后的日志及描述如图 5.5 所示。

```
commit 808f776d5775fba14495abbef941a66116ff4418
Author: yan-liao-wenjin <2740429059@qq.com>
Date: Tue May 28 14:37:01 2024 +0800

    相较于上一次，我通过CSS修改了网页的样式。* 选择器用于选择页面上的所有元素，并为它们设置了默认的外边距（margin）和文本对齐方式（text-align）。这种全局设置通常用于确保页面元素之间有适当的间距，并且文本内容居中显示。使用了JavaScript，这段代码定义了一个自定义的 $ 函数，该函数旨在简化对DOM元素的选择过程，同时设置了两个事件监听器以捕获用户在页面上特定元素（具有标识符bookface）上的鼠标行为。
```

图 5.4 第三次迭代代码提交日志

6. 适用移动互联时代的响应式设计

6.1 分析和设计

在数字化时代，适应不同设备屏幕的响应式网页设计变得极为关键。随着移动设备成为人们连接网络的主要方式，一个能够灵活调整以适应各种屏幕大小和分辨率的网站，对于提供卓越的用户体验至关重要。这种设计确保了网站在不同设备上保持统一的外观和操作体验，使用户能够无缝地浏览内容，进而提升满意度和忠诚度。响应式设计还能显著减少网站的维护开销。传统网站设计往往需要为不同设备定制多个版本，这不仅增加了开发复杂性，还增加了维护和更新的成本。而响应式设计通过单一的代码库适应多种设备，有效降低了这些成本，同时实现了网站内容的即时同步更新，提高了维护的效率。从 SEO 的视角来看，响

应式设计同样具有显著优势。搜索引擎更倾向于提升那些提供优秀移动体验的网站排名。因此，采用响应式设计有助于网站在移动搜索结果中获得更优的位置，吸引更多的潜在用户。同时，响应式设计还能加快网站的加载速度。通过优化图像和其他资源，减少了页面的体积和加载时间，提升了网页的访问速度和性能，这对于改善用户体验和减少用户流失至关重要。

响应式设计对于加强代码的一致性也同样重要。它通过在不同设备上提供一致的用户体验，帮助建立用户对品牌的信任和忠诚度。这种一致性不仅体现在视觉设计上，还包括操作体验和内容展示，有助于提升品牌的整体形象和市场竞争能力。其中，第四次迭代的用例图如图 6.1 所示。

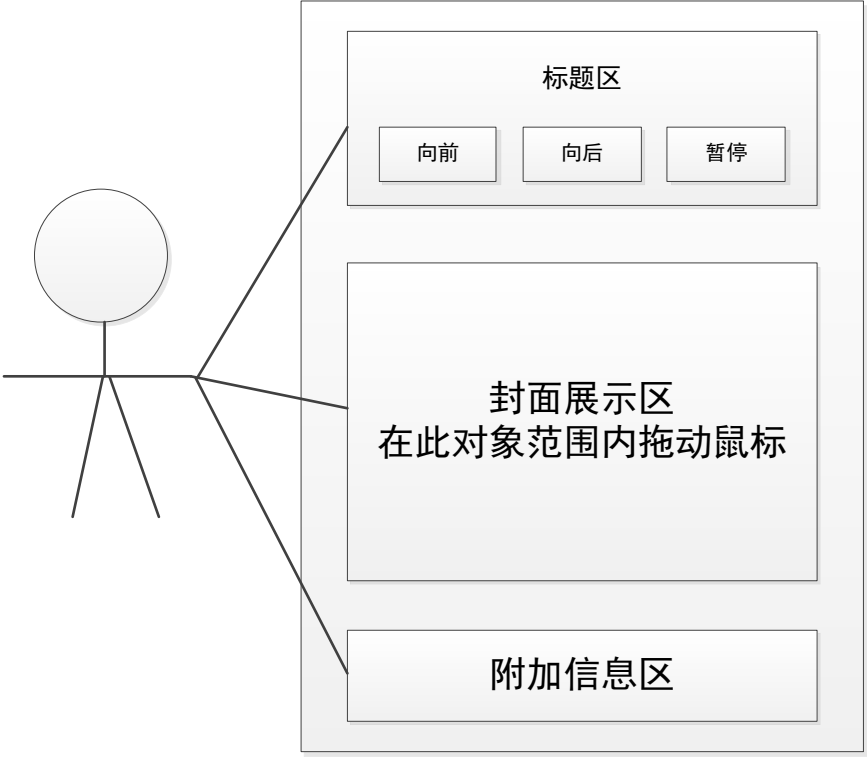


图 6.1 第四次迭代用例图

6.2 项目的实现和编程

在这次改进中，除了捕捉鼠标点击的操作，还捕捉了鼠标拖动的操作，代码的改进集中在 JavaScript 的部分上。包括响应式布局设置、自定义 DOM 元素选择函数以及鼠标事件监听。

代码定义了一个名为 UI 的空对象，用于存储与界面相关的尺寸信息。通过判断 `window.innerWidth`（浏览器窗口的宽度）是否大于 600 像素，来决定

UI.appWidth（应用程序的宽度）的值。如果窗口宽度大于 600，则 UI.appWidth 设置为 600；否则，UI.appWidth 与窗口宽度相同。接着，我将 UI.appHeight 设置为 window.innerHeight（浏览器窗口的高度）。根据 UI.appWidth 计算出基础字体大小 baseFont，并将其设置为 document.body 的字体大小。这样做的好处是可以通过改变 body 对象的字体大小来影响其后代的字体大小，从而在一定程度上实现响应式设计。

同时设置 document.body 的宽度和高度。宽度设置为 UI.appWidth，而高度则是窗口高度减去 62 像素（为了留出一些空间给浏览器工具栏或其他界面元素）。通过一个条件判断来隐藏或显示某个元素（其 ID 或选择器为"aid"）。如果窗口宽度小于 1000 像素，则将该元素隐藏。这样可以实现无论窗口宽度如何，都设置了该元素的宽度和高度。元素的宽度是窗口宽度减去 UI.appWidth 再减去 30 像素，高度则是 UI.appHeight 减去 62 像素。

接下来实现了对鼠标事件的监听。首先定义了一个名为 mouse 的空对象，用于存储与鼠标相关的状态信息，如 isDown（鼠标是否按下）、x（鼠标的 X 坐标）和 deltaX（鼠标在 X 轴上的位移）。然后，为 ID 或选择器为"bookface"的元素添加了 mousedown 和 mousemove 事件监听器。当鼠标在该元素上按下或移动时，会打印出鼠标的坐标，并更新该元素的文本内容为鼠标的坐标。

```
<script>
  var UI = {};
  if(window.innerWidth>600){
    UI.appWidth=600;
  }else{
    UI.appWidth = window.innerWidth;
  }

  UI.appHeight = window.innerHeight;

  let baseFont = UI.appWidth /20;
  //通过改变 body 对象的字体大小，这个属性可以影响其后代
  document.body.style.fontSize = baseFont + "px";
  //通过把 body 的高度设置为设备屏幕的高度，从而实现纵向全屏
  //通过 CSS 对子对象百分比（纵向）的配合，从而达到我们响应式设计的目标
  document.body.style.width = UI.appWidth - baseFont + "px";
  document.body.style.height = UI.appHeight - baseFont*4 + "px";
  if(window.innerWidth<1000){
```

```

        $("aid").style.display='none';
    }
    $("aid").style.width=window.innerWidth-UI.appWidth - baseFont*3
    +'px';
    $("aid").style.height= UI.appHeight - baseFont*3 +'px';

    //尝试对鼠标设计 UI 控制
    var mouse={};
    mouse.isDown= false;
    mouse.x= 0;
    mouse.y= 0;
    mouse.deltaX=0;
    $("bookface").addEventListener("mousedown",function(ev){
        mouse.isDown=true;
        mouse.x= ev.pageX;
        mouse.y= ev.pageY;
        console.log("mouseDown at x: "+(" "+mouse.x +", " +mouse.y +")" );
        $("bookface").textContent= "鼠标按下，坐标：
    "+"(" +mouse.x+", " +mouse.y+")";
    });
    $("bookface").addEventListener("mouseup",function(ev){
        mouse.isDown=false;

        $("bookface").textContent= "鼠标松开!";
        if(Math.abs(mouse.deltaX) > 100){
            $("bookface").textContent += "，这是有效拖动！ " ;
        }else{
            $("bookface").textContent += " 本次算无效拖动！ " ;
            $("bookface").style.left = '7%' ;
        }
    });
    $("bookface").addEventListener("mouseout",function(ev){
        ev.preventDefault();
        mouse.isDown=false;

        $("bookface").textContent= "鼠标松开!";
        if(Math.abs(mouse.deltaX) > 100){
            $("bookface").textContent += " 这次是有效拖动！ " ;
        }else{
            $("bookface").textContent += " 本次算无效拖动！ " ;
            $("bookface").style.left = '7%' ;
        }
    }

```

```

});
$("#bookface").addEventListener("mousemove",function(ev){
    ev.preventDefault();
    if (mouse.isDown){
        console.log("mouse isDown and moving");
        mouse.deltaX = parseInt( ev.pageX - mouse.x );
        $("#bookface").textContent= "正在拖动鼠标，距离： " + mouse.deltaX
        +"px 。 ";
        $('#bookface').style.left = mouse.deltaX + 'px' ;
    }
});

```

```

function $(ele){
    if (typeof ele !== 'string'){
        throw("自定义的$函数参数的数据类型错误，实参必须是字符串!");
        return
    }
    let dom = document.getElementById(ele) ;
    if(dom){
        return dom ;
    }else{
        dom = document.querySelector(ele) ;
        if (dom) {
            return dom ;
        }else{
            throw("执行$函数未能在页面上获取任何元素，请自查问题!");
        }
        return ;
    }
}
} //end of $

```

</script>

在最后，代码定义了一个自定义的 DOM 元素选择函数\$。该函数接受一个字符串参数 ele，用于指定要选择的元素的 ID 或 CSS 选择器。函数首先检查参数的数据类型是否为字符串，如果不是则抛出错误。函数使用 getElementById 和 querySelector 方法选择元素。如果找到了元素，则返回该元素；否则，抛出错误提示未能在页面上获取任何元素，提高了代码的复用性和可读性。

6.3项目的运行和测试

该页面的二维码和实际效果如图 6.2，图 6.3 和图 6.4 所示。

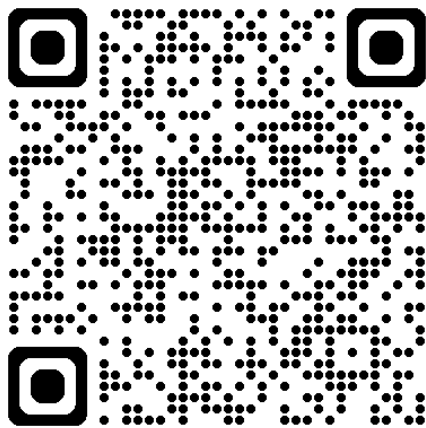


图 6.2 第四次迭代代码网页二维码图



6.3 第四次迭代代码手机端效果图

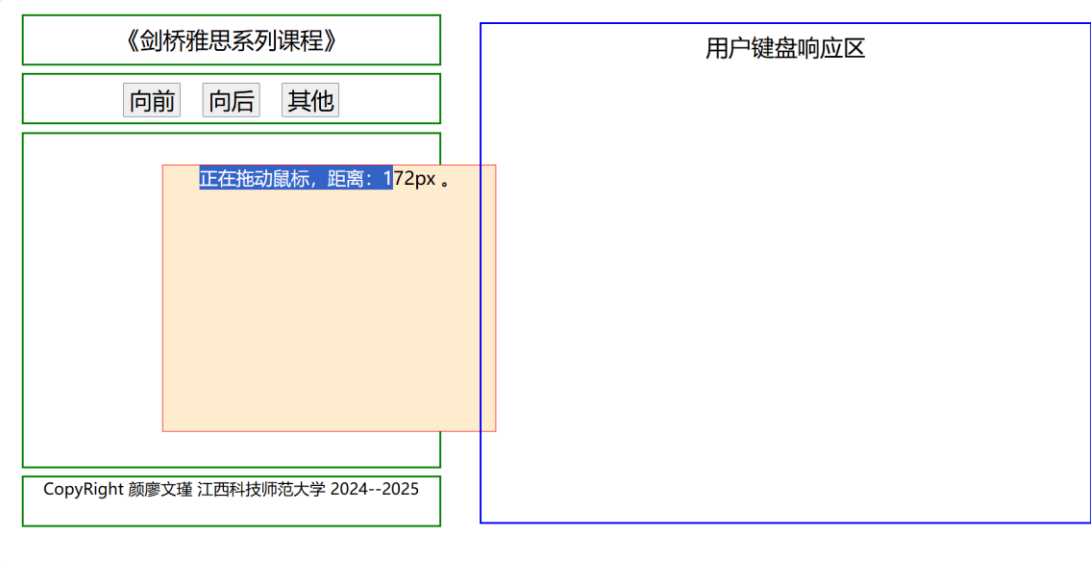


图 6.4 第四次迭代 PC 端网页实际效果

6.4项目的代码提交和版本管理

第四次代码提交的日志如图 6.5 所示。

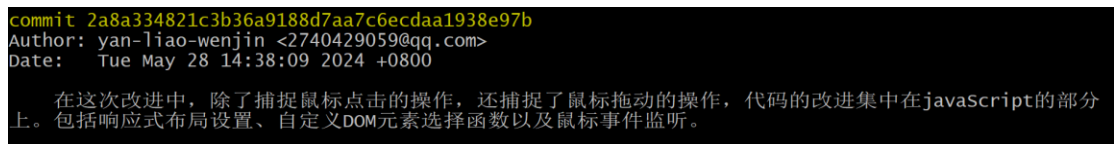


图 6.5 第四次迭代代码提交日志

7. 个性化 UI 设计中鼠标模型以及通用 UI 设计

7.1 分析和设计

个性化用户界面设计在现代软件与网页设计领域扮演着核心角色。随着用户界面复杂度的提升以及市场竞争的加剧，一个既独特又满足用户需求的个性化用户界面设计，对提升产品吸引力及增强用户满意度和忠诚度至关重要。

个性化用户界面设计能够彰显产品的独特性和品牌形象。精心设计的界面元素、色彩搭配与交互方式，使产品在众多竞争者中独树一帜，给用户留下深刻印象。这种独特性不仅增强了产品识别度，也加深了用户对品牌的认知与记忆。个性化用户界面设计亦能提升用户体验。符合用户习惯和喜好的界面设计降低了用户的学习成本，使用户能够更轻松、高效地完成任务。通过深入理解用户需求和行为，设计师能够创造出更人性化、直观的界面，从而提高用户满意度和忠诚度。

个性化用户界面设计增强产品的可访问性和包容性。通过优化字体大小、颜色对比度、键盘操作等细节，设计师确保产品能够满足不同用户群体的需求，包括视力不佳、色弱、行动不便等用户。这种包容性设计扩大了产品的受众范围，增强了产品的市场竞争力。个性化用户界面设计促进产品的创新与发展。设计师通过不断尝试新的设计理念和交互方式，推动产品在功能、视觉和体验方面的持续进步。这种创新精神不仅满足了用户日益增长的需求，也为产品带来了新的增长点和发展机遇。

个性化用户界面设计在提升产品吸引力、增强用户体验、增强产品可访问性和包容性以及促进产品创新和发展等方面发挥着重要作用。在设计和开发产品时，重视个性化用户界面设计显得极为必要。这次代码的用例图如图 7.1 所示。

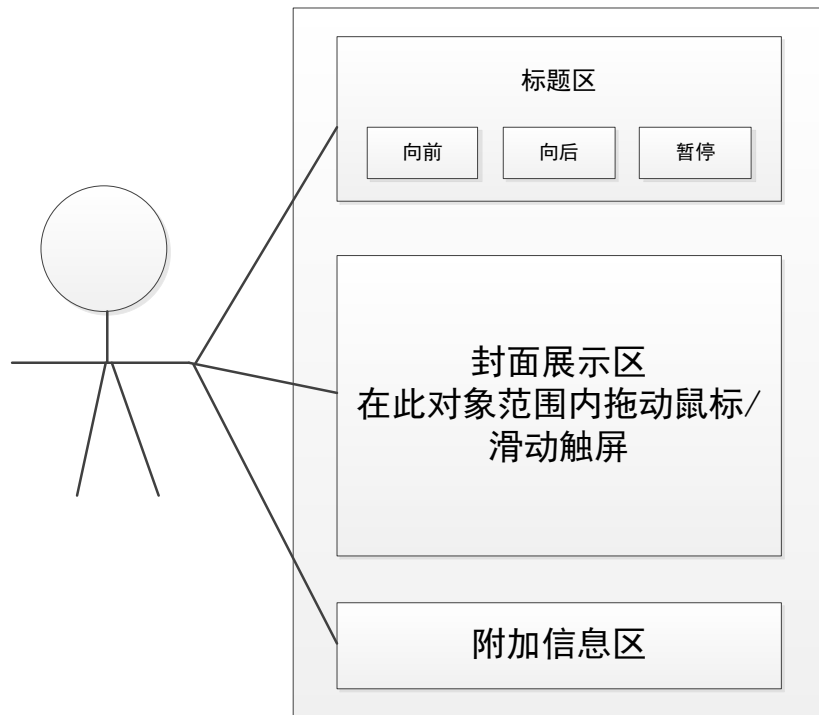


图 7.1 第五次迭代用例图

7.2 项目的实现和编程

在上次代码的基础上，增加了个性化 UI 的设计。代码段定义了一个名为 **Pointer** 的对象，其功能是管理鼠标和触屏的交互事件。在初始化过程中，**Pointer** 对象配置了三个属性：**isDown**，用于监控鼠标或触屏是否被按下；**x**，记录上一次交互的 **x** 坐标；**deltaX**，用于计算滑动或拖动的位移。随后，定义了三个事件处理函数：**handleBegin**、**handleEnd** 和 **handleMoving**，分别对应交互的起始、终止和移动事件。

handleBegin 函数在鼠标按下或触屏接触时触发，将 **Pointer.isDown** 属性设置为 **true**，并根据事件类型记录相应的坐标。随后，更新页面中具有特定 ID（"bookface"）的元素的文本，以反映事件类型和坐标。**handleEnd** 函数在鼠标释放或触屏接触结束时执行，将 **Pointer.isDown** 属性设置为 **false**，并阻止事件的默认行为。然后，根据触屏事件的发生以及滑动距离，判断滑动的有效性，并更新 "bookface" 元素的文本和样式。

handleMoving 函数处理鼠标移动或触屏滑动事件。它首先阻止事件的默认行为，然后检查是否为触屏事件且 **Pointer.isDown** 为 **true**。满足条件时，计算滑动

距离并更新"bookface"元素的文本和样式。对于鼠标事件，执行相似的处理逻辑。接着通过自定义的\$函数实现对页面上特定元素（假定其 ID 为"bookface"）的事件监听器绑定，用以处理鼠标和触屏的交互操作。这些事件监听器覆盖了从鼠标按下和触屏接触开始，到鼠标释放、触屏接触结束、鼠标移出以及鼠标和触屏移动整个流程。触发这些事件时，将调用 `handleBegin`、`handleEnd` 和 `handleMoving` 等相应的处理函数，以实现用户对交互的灵活响应。

我还为整个 `body` 元素绑定了键盘按键事件监听器。用户在页面上按下键盘按键时，会触发一个匿名函数，该函数将按键的字符值追加到 ID 为"aid"的元素的文本内容中，从而实现键盘输入的简单处理。

自定义的\$函数作为 DOM 选择器的简化版本，接受一个字符串参数——预期为元素的 ID 或 CSS 选择器，并尝试在页面中查找匹配的 DOM 元素。找到元素时返回该元素，否则抛出错误提示。这种设计不仅提升了代码的可读性和可维护性，还增强了对用户交互的灵活和精细控制。

7.3项目的运行和测试

代码的二维码和实际效果如图 7.2，图 7.3 和图 7.4 所示

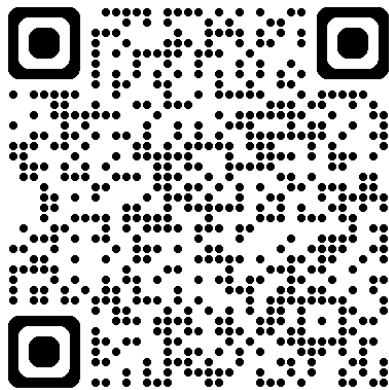


图 7.2 第五次迭代网页二维码



图 7.3 第五次迭代手机端效果

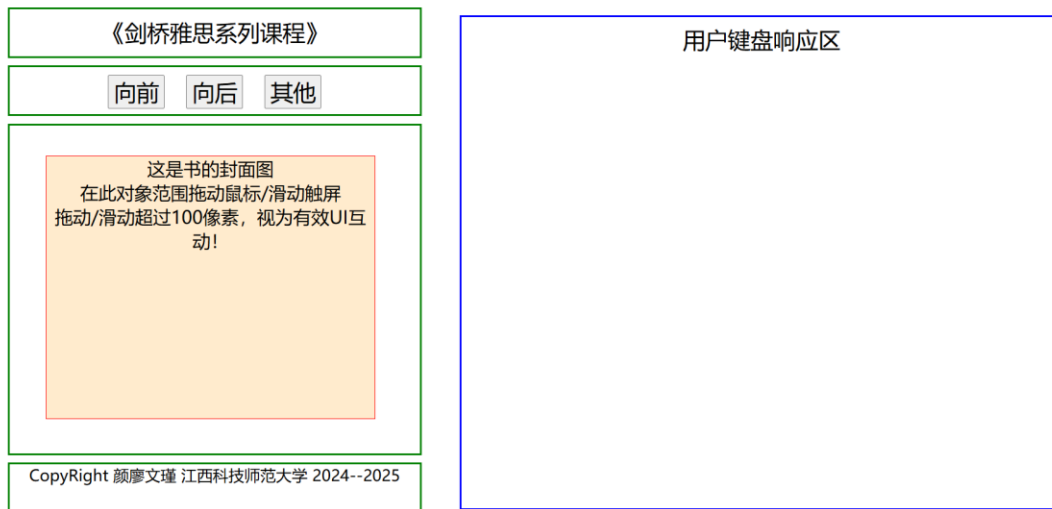


图 7.4 第五次迭代 PC 端网页实际效果

7.4项目的代码提交和版本管理

本次代码提交后的日志如图 7.5 所示。

```
commit 8bc06f06b6097caa5d2149b1153a1f2417f868ff
Author: yan-liao-wenjin <2740429059@qq.com>
Date: Tue May 28 14:38:58 2024 +0800

    在上次代码的基础上，增加了个性化UI的设计。代码段定义了一个名为Pointer的对象，其功能是管理鼠标和触屏的交互事件。在初始化过程中，Pointer对象配置了三个属性：isDown，用于监控鼠标或触屏是否被按下；x，记录上一次交互的x坐标；deltaX，用于计算滑动或拖动的位移。随后，定义了三个事件处理函数：handleBegin、handleEnd和handleMoving，分别对应交互的起始、终止和移动事件。
```

图 7.5 第五次迭代代码提交日志

8. UI 的个性化键盘控制

8.1分析和设计

在这个阶段，我应用 keydown 和 keyup 键盘底层事件，为未来的键盘功能提供更大空间。代码段首先为 body 元素配置了两个事件监听器，分别用于处理键盘按键按下（keydown）和按键松开（keyup）的操作。当键盘上的任意键被按下时，将触发 keydown 事件，并执行相应的回调函数。在此回调函数中，首先调用 `ev.preventDefault()` 以阻止事件的默认行为，随后捕获按键的键值（`ev.key`）和字符编码（`ev.keyCode`）。然后，利用自定义的 `$` 函数定位页面上 ID 为 keyboard

的元素，并更新其文本内容以显示按下的键及其字符编码。这次迭代的用例图如图 8.1 所示。

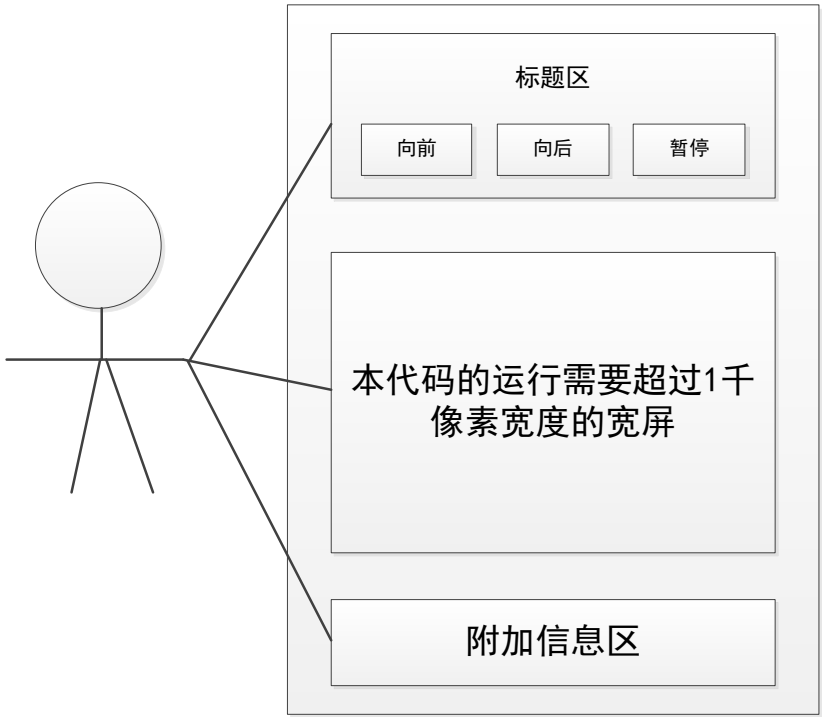


图 8.1 第六次迭代用例图

8.2 项目的实现和编程

键盘按键松开时，触发 `keyup` 事件，并执行对应的回调函数。该函数同样首先阻止事件的默认行为，捕获按键的键值和字符编码。接着，更新 `keyboard` 元素的文本内容，以显示松开的键及其字符编码。此外，回调函数还调用了 `printLetter` 函数，用以判断按键是否应当被打印到其他元素中。若 `printLetter` 函数返回 `true`，则将按键的键值追加至 ID 为 `typeText` 的元素的文本内容。

`printLetter` 函数负责确定传入的键值是否应当被打印。它首先验证键值的长度是否大于 1，这通常用于识别组合键或特殊键（如 `Ctrl`、`Shift` 等），若条件满足，则返回 `false`，表示不进行打印。然后，定义了一个包含常见标点符号的数组 `puncs`。函数进一步检查键值是否为字母、数字或标点符号集合中的成员，若是，则返回 `true`，表示应当打印；否则，返回 `false`。

代码中还定义了一个名为 `$` 的函数，用于根据传入的字符串参数在页面上查找 DOM 元素。该函数首先验证参数是否为字符串类型，若不是，则抛出错误。

接着，尝试通过 `getElementById` 方法定位元素，若未找到，则使用 `querySelector` 方法继续尝试。若最终未能找到元素，则抛出错误提示；若找到，则返回相应的 DOM 元素。

//提出问题：研究利用"keydown"和"keyup"2 个底层事件，实现同时输出按键状态和文本内容

```
    $("body").addEventListener("keydown",function(ev){
    ev.preventDefault() ;
    let k = ev.key;
    let c = ev.keyCode;
    $("keyboard").textContent = "您已按键 : " + k + " , "+ "字符编码 : " + c;
    });
    $("body").addEventListener("keyup",function(ev){
    ev.preventDefault() ;
    let key = ev.key;
    let code = ev.keyCode;
    $("keyboard").textContent = "松开按键 : " + key + " , "+ "字符编码 : " + code;
    if (printLetter(key)){
        $("typeText").textContent += key ;
    }
    function printLetter(k){
    if (k.length > 1){ //学生必须研究这个逻辑的作用
        return false ;
    }
    let puncs =
    ['~','`','!','@','#','$','%','^','&','*','(',')','_','+','=',' ','.', '<','>','?','/',' '];
        if ( (k >= 'a' && k <= 'z') || (k >= 'A' && k <= 'Z') || (k >= '0' && k <= '9')) {
            console.log("letters") ;
            return true ;
        }
        for (let p of puncs ){
            if (p === k) {
                console.log("puncs") ;
                return true ;
            }
        }
        return false ;
    }
    //提出更高阶的问题，如何处理连续空格和制表键 tab?
} //function printLetter(k)
});
```

```

    } //Code Block End
function $(ele){
    if (typeof ele !== 'string'){
        throw("自定义的$函数参数的数据类型错误, 实参必须是字符串!");
        return
    }
    let dom = document.getElementById(ele) ;
    if(dom){
        return dom ;
    }else{
        dom = document.querySelector(ele) ;
        if (dom) {
            return dom ;
        }else{
            throw("执行$函数未能在页面上获取任何元素, 请自查问题!");
        }
        return ;
    }
}
} //end of $

```

8.3项目的运行和测试

该页面的在 PC 端上的最终效果如图 8.4 所示, 在手机端的效果如图 8.3 所示, 网页的二维码如图 8.2 所示。

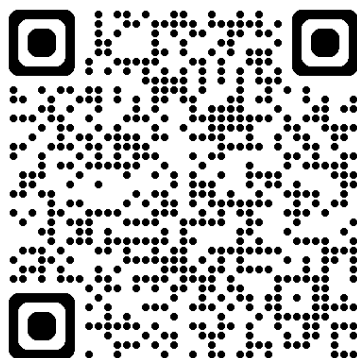


图 8.2 第六次迭代二维码



图 8.3 第六次迭代手机端网页实际效果

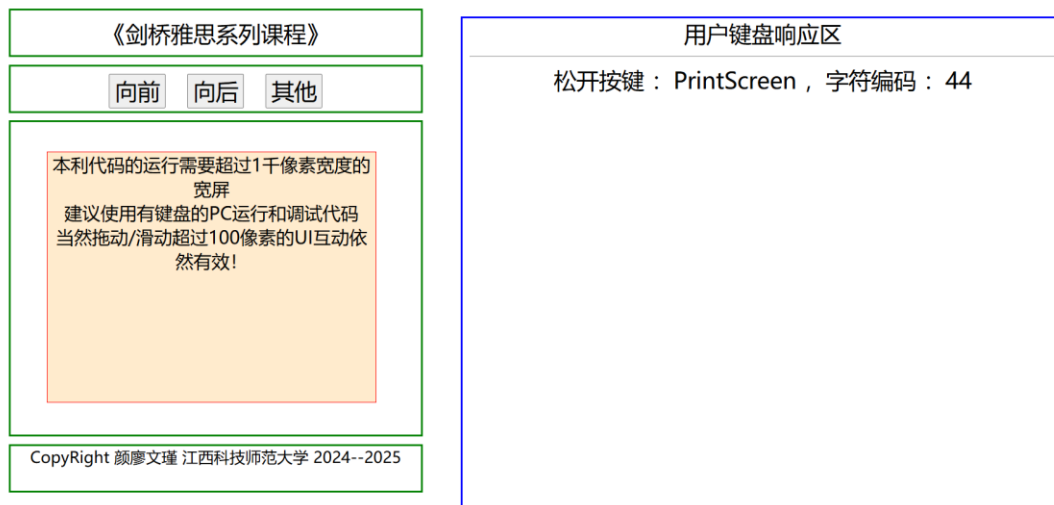


图 8.4 第六次迭代 PC 端网页实际效果

8.4 项目的代码提交和版本管理

第六次提交代码的日志如图 8.5 所示。

```
commit b0ca1de6235ee35b5aba8ea91abbabfb4251d630 (HEAD -> main)
Author: yan-liao-wenjin <2740429059@qq.com>
Date: Tue May 28 14:40:25 2024 +0800

    应用keydown和keyup键盘底层事件，为未来的键盘功能提供更大空间
```

图 8.5 第六次迭代代码提交日志

9. 本项目中的高质量代码

在当代，计算机的普及程度已与螺丝刀相媲美，然而它们的复杂性却远非螺丝刀所能比拟。要让计算机按照个人的意愿执行任务，并非总是轻而易举。

对于那些常见且易于理解的计算机任务，例如查看电子邮件或进行计算，用户可以简单地启动相应的应用程序并立即着手工作。然而，对于那些独特或开放式的任务，可能并不存在现成的应用程序来满足需求。

在这种情况下，编程便显得尤为重要。编程，即编写程序的过程，涉及创建一系列精确的指令，指导计算机执行特定的操作。鉴于计算机的机械性和不灵活的特性，编程本质上可能显得单调且令人挫败。不过，若能超越这些挑战，甚至可能从中发现乐趣，享受与这些不灵活的机器进行严谨思维的过程，编程便能带

来丰厚的回报。它使得人们能够在转瞬之间完成那些手动操作需要耗费大量时间的任务。此外，编程还是一种极佳的抽象思维训练方式，能够极大地拓展计算机工具的功能，使其能够执行以往无法完成的任务^[6]。

在构建一个同时支持鼠标和触屏控制的系统时，我采用了 MVC（Model-View-Controller）设计模式来确保应用程序的逻辑、数据和用户界面之间的清晰分离。首先，我们需要定义一个 `PointerModel`，它负责管理指针（可能是光标或虚拟手指）的当前位置、状态（如按下、移动、释放）等信息，并提供更新这些信息的接口。接着设计一个 `PointerView`，它负责根据 `Model` 提供的数据在屏幕上显示指针，并包含必要的绘图逻辑。最后，我们实现一个 `PointerController`，它负责监听鼠标和触屏事件，当这些事件发生时，更新 `Model` 中的指针状态，并通过 `Model` 触发的事件来通知 `View` 进行必要的更新。

在集成 MVC 的过程中，我首先在应用程序启动时创建并初始化 `Model`、`View` 和 `Controller` 的实例。然后将 `Controller` 的事件监听器绑定到鼠标和触屏事件上，以确保当这些事件发生时能够正确地更新 `Model` 和 `View`。当 `Controller` 接收到鼠标或触屏事件时，它会提取必要的信息，如位置和状态，并更新 `Model`。`Model` 在状态改变时会触发相应的事件，并将新的状态传递给注册的事件监听器，包括 `View` 的监听器。`View` 在收到这些事件时，会根据 `Model` 提供的新数据更新其显示。这种 MVC 设计模式的应用不仅保证了应用程序的解耦性，提高了可维护性和可扩展性，还使得整个系统更加易于测试和调试。如果需要支持新的输入设备或添加新的功能，只需修改或扩展 `Controller` 和/或 `Model`，而无需更改 `View`，从而大大降低了代码的复杂性和耦合度。

10. 用 gitBash 工具管理项目的代码仓库和 http 服务器

10.1 经典 Bash 工具介绍

在讨论命令行界面时，我们通常指的是 `Shell`。`Shell` 是一个软件程序，它负责接收用户通过键盘输入的指令，并将其传递给操作系统以执行相应的操作。大多数 Linux 发行版都配备了 GNU 项目开发的 `Shell` 程序，称为 `bash`。“`bash`”是

"Bourne Again Shell"的缩写，它是基于 Steve Bourne 开发的原始 Unix Shell 程序 "sh"的改进版本。

与 Windows 操作系统类似，类 Unix 操作系统如 Linux 采用层次化的目录结构来组织文件。这种结构将文件以树状模式组织，其中目录（在某些系统中也称为文件夹）可以包含文件和其他目录。在文件系统的最顶层是根目录，它包含文件和子目录，而这些子目录又可以包含更多的文件和子目录，形成一个递归的层次结构^[7]。

10.2 通过 github 平台访问项目

借助 github 网站的图形化上传文件界面将本项目中的代码上传至 github 后，可以进入 <https://iris2287.github.io/>地址来访问，最终项目的页面如图 10.1 所示。



图 10.1 项目最终效果图

参考文献

- [1] W3C.W3C's history. W3C Community. [EB/OL].<https://www.w3.org/about/>.<https://www.w3.org/about/history/>. 2023.12.20
- [2] Douglas E. Comer. The Internet Book [M] (Fifth Edition). CRC Press Taylor & Francis Group, 2019: 217-218
- [3] John Dean, PhD. Web programming with HTML5,CSS, and JavaScript[M].Jones & Bartlett Learning, LLC.2019: 2
- [4] John Dean, PhD. Web programming with HTML5,CSS,and JavaScript[M].Jones & Bartlett Learning, LLC. 2019: xi
- [5] Behrouz Forouzan. Foundations of Computer Science[M](4th Edition). Cengage Learning EMEA,2018: 274--275
- [6] Marijn Haverbeke. Eloquent JavaScript 3rd edition. No Starch Press,Inc, 2019.
- [7] William Shotts. The Linux Command Line, 2nd Edition [M]. No Starch Press, Inc, 245 8th Street, San Francisco, CA 94103, 2019: 3-7

Web APP UI Development Technologies

Abstract: Over the past decade, HTML5, as the core of Web standards, has gained extensive recognition and application in the field of software development due to its cross-platform capabilities and open-source nature. Relying on Web client technology, this research project has deeply explored and practiced program design and software development. Through in-depth research on technical documents and participation in discussions in the developer community, we have successfully designed and implemented an application with a personalized user interface. In the development process, we directly utilized the fundamental APIs of Web clients, and employed HTML, CSS, and JavaScript technologies to construct content, design interfaces, and implement interactive functions, respectively. Additionally, the project implemented a responsive design strategy to meet the diverse device requirements in the mobile internet era. It also adopted an object-oriented programming approach, constructing a pointer model through a unified codebase, achieving compatibility with different input devices, thereby enhancing code quality and maintainability. In terms of project management, an incremental development method was adopted, with multiple iterations (including analysis, design, implementation, and testing) to gradually optimize and perfect the product. Regarding open-source and code sharing, Git was used for version control. After several code reconstructions and corrections, the project code was finally published on the GitHub platform. Leveraging the HTTP services provided by GitHub, we deployed the application globally, enabling users to access it easily through web links or scanning QR codes, achieving cross-platform accessibility.

Key Words: Computer Science and Technology; Web Front-end Development; HTML; CSS; Git;