

成绩:

# 江西科技师范大学

## 课程论文

题目(中文): 基于 Web 客户端技术个性化 UI 设计与实现

(外文): Web client based customized UI design and  
Programming

院系: 元宇宙产业学院

专业: 计算机科学与技术

学生姓名: 彭锦鹏

学号: 20213625

指导教师: 李健宏

2024 年 6 月 16 日

# 目录

1. 前言 .....	1
1.1 毕业设计任务分析 .....	1
1.2 研学计划 .....	2
1.3 研究方法 .....	2
2. 运用技术与编译环境介绍 .....	3
2.1 HTML .....	3
2.2 CSS .....	4
2.3 Java Script .....	4
2.4 Git Bash 代码管理 .....	4
2.5 Visual Studio Code .....	5
3. 增量式迭代开发模式 (ADIT) .....	5
4. “三段论”式的内容设计与实现 .....	6
4.1 分析与设计 .....	6
4.1.1. 分析 .....	6
4.1.2. 设计 .....	7
4.2 代码实现 .....	8
4.2.1 HTML 代码 .....	8
4.2.2 CSS 代码 .....	8
4.3 运行与测试 .....	9
5. 响应式设计与适应窄屏实现 .....	10
5.1 分析与设计 .....	10
5.1.1. 分析 .....	10
5.1.2 设计 .....	11
5.2 代码实现 .....	12
5.2.1 HTML 代码 .....	12
5.2.2 CSS 代码 .....	12
5.2.3 JavaScript 代码 .....	14
5.3 运行与测试 .....	14
6. 个性化交互 UI 设计——鼠标模型 .....	16
6.1 分析与设计 .....	16
6.1.1 分析 .....	16
6.1.2 设计 .....	16
6.2 代码实现 .....	17
6.2.1 HTML 代码 .....	17
6.2.2 CSS 代码 .....	17
6.2.3 JavaScript 代码 .....	18
6.3 运行与测试 .....	20
7. 通用 UI 设计为触屏和鼠标统一建模 .....	21
7.1 分析与设计 .....	22
7.1.1 分析 .....	22

## 目录

7.1.2 设计 .....	22
7.2 代码实现 .....	22
7.2.1 HTML 代码 .....	22
7.2.2 CSS 代码 .....	23
7.2.3 JavaScript 代码 .....	25
7.3 运行与测试 .....	28
8. 个性化 UI 监控键盘实现 .....	29
8.1 分析与设计 .....	29
8.1.1 分析 .....	29
8.1.2 设计 .....	30
8.2 代码实现 .....	30
8.2.1 HTML 代码 .....	30
8.2.2 CSS 代码 .....	31
8.2.3 JavaScript 代码 .....	33
8.3 运行与测试 .....	35
8.4 代码提交与版本控制 .....	36
9. 用 Git Bash 工具管理项目的代码仓库和 Http 服务器 .....	36
9.1 跨世纪的经典 Bash 工具 .....	36
9.2 通过 Git Hub 平台实现本项目的全球域名。 .....	37
参考文献 .....	39

# 基于 Web 客户端技术个性化 UI 设计与实现

**摘要：**HTML5 作为 Web 标准开发的核心技术，具有平台兼容性强、开源等优势，在各个领域得到广泛应用。致力于利用 HTML5 等前端技术开发个性化 UI 应用程序。通过查阅资料和论坛，进行深入研究并综合运用 HTML5 进行内容建模、CSS 进行 UI 设计、JavaScript 实现交互功能。项目采用响应式设计，智能适应不同屏幕需求。应用面向对象编程思想，编写通用指针模型控制用户输入，确保代码高质量。工程采用增量式开发，通过六次 ADIT 迭代完善项目并进行测试。在代码管理方面，使用 Git 进行版本管理，进行正式提交和两次修改提交。最终，通过 Git Bash 上传至知名 GitHub，借助其 HTTP 服务器实现全球范围内应用程序高效部署和访问。这个项目不仅展示了我们的技术实力和创新能力，也提升了团队合作与项目管理水平。在未来，我们将继续研究和应用 HTML5 等前端技术，开发更出色的 Web 应用，为用户提供更优质的服务体验，并继续推动技术的分享与发展。

**关键字：**Web 前端;JavaScript; GitHub; Git-Bash。

## 1. 前言

### 1.1 毕业设计任务分析

毕业设计是大学生在学科专业基础上进行的一项科研项目，旨在提升学生的综合能力，培养其解决实际问题的能力。毕业设计任务分析是指对毕业设计题目和要求进行深入分析，明确任务的目标和要求，为设计过程提供指导和依据。

首先，在进行毕业设计任务分析时，需要对设计题目进行详细的解读和分析。了解设计题目的背景、意义和目的，明确设计要解决的问题和研究的重点，有助于确定研究方向和方法。

其次，需要分析任务的要求和限制条件。毕业设计通常有一定的要求和限制条件，如设计范围、时间要求、研究方法等。通过分析这些要求和条件，可以确定设计的可行性和实施方案，避免出现不可实现或违反规定的情况。

此外，还需对设计任务的目标和成果进行明确定义。毕业设计是为了实现一定的研究目标和取得一定的成果，如提出新的理论、方法或技术，解决实际问题，推进学科发展。因此，需要明确设计的目标和预期成果，为设计过程和评价提供依据。

最后，需要对设计任务进行综合分析和评估，确定具体的研究方向和方法。

根据任务分析的结果，可以确定研究方向和实施步骤，制定详细的研究计划，包括研究内容、方法、时间表和任务分工等。同时，还需评估设计任务的难度和重要性，确定研究的深度和广度，以保证设计的质量和价值<sup>[1]</sup>。

综上所述，毕业设计任务分析是毕业设计过程中至关重要的一环，通过深入分析设计题目、要求和限制条件，明确设计的目标和成果，确定研究方向和方法，为设计过程提供指导和依据，确保设计的顺利完成和成果的取得。

## 1.2 研学计划

研学计划是指根据学科专业要求，结合学生个人兴趣和发展需求，通过实地考察、调研和实践活动，进行探究性学习和跨学科研究的计划。研学计划的目的是促进学生全面发展和提高学习效果，培养学生的独立思考和创新能力。通过制定研学主题、确定时间和地点、安排具体活动和任务、组建团队和指导老师、评估和总结效果和成果等步骤，学生可以在实践中不断探索和学习，提升自身的综合素质和能力。研学计划有助于拓展学生的视野、加深对知识的理解，激发学生的学习兴趣和潜力，为其未来的发展打下坚实的基础<sup>[2]</sup>。

## 1.3 研究方法

本研究计划旨在借助文献法和模型研究法，探索大数据分析在商业决策中的应用。通过深入搜集文献资料和建立数据分析模型，本研究将对大数据分析和商业决策进行系统性研究和分析，旨在为企业提供更有效的决策支持 and 应用指导。首先，将通过文献法深入了解大数据分析、商业决策和研究方法的相关理论和实践经验，以建立研究框架和理论基础。其次，将利用模型研究法建立数据分析模型，并结合实际商业案例数据进行模拟和分析，探讨文献法和模型研究法在商业决策中的应用效果和潜在贡献。研究过程中，将结合理论分析和实证研究相结合，深入挖掘文献法和模型研究法在大数据分析和商业决策中的优势和局限，旨在为企业提供更科学、准确的决策支持和实践指导。最终，通过撰写研究报告与实践指南，总结研究发现并提出对企业的实际价值和实施建议，推动企业在大数据时代的商业决策过程中迈出更加坚实、可靠的步伐，为企业的可持续发展和竞争力提升贡献力量。通过该研究计划的实施，希望能够提升企业在大数据时代的决策能力，开拓新的商业前景，助力企业实现更加可持续、创新和成功的发展。

## 2. 运用技术与编译环境介绍

本章节介绍了 Web UI 系统实现过程的现实背景，并描述了在系统开发过程中使用的关键技术。我们将探讨构建现代 Web 应用的基础技术，包括 HTML、CSS 和 JavaScript，这些技术共同构成了网页的核心。此外，还将介绍 Git-bash 这一在版本控制和代码管理中必不可少的工具。为了支持高效的开发流程，本章还将介绍所使用的开发环境，该环境为开发者提供了稳定且高效的工作空间。在编译器选择方面，我们将讨论 Visual Studio Code，这是一款功能强大、备受推崇的代码编辑器，其丰富的特性、灵活的扩展和高效的代码处理能力，使其成为开发者的首选工具。

### 2.1 HTML

HTML（超文本标记语言）是构建互联网网页和应用的基础语言，通过标签和属性定义网页的结构和内容，使开发者能够以结构化的方式组织网页中的各种元素，如文本、图片、链接、表格和列表等。HTML 支持创建超链接，这是互联网相互连接的核心特征，允许网页之间进行跳转。此外，HTML 还可以集成多媒体内容，如视频和音频，并通过表单来收集用户输入。虽然 HTML 本身不负责样式设计，但与 CSS 和 JavaScript 结合使用，可以精确控制网页的布局、外观和行为，为开发者提供强大的工具。HTML 作为一种标准化语言，确保不同浏览器和设备上的网页呈现一致，促进了 Web 开发技术的广泛传播和应用。随着 HTML 的不断进步，如 HTML5 引入的新特性和改进，进一步推动了 Web 技术的发展，满足现代 Web 应用的多样化需求，包括响应式设计、移动优先策略和无障碍访问。总体而言，HTML 不仅支撑着全球信息的自由流通，也是推动互联网创新和经济增长的关键因素。HTML 具有以下主要特点：首先，它通过各种标签来区分内容的结构，例如标题、段落、链接等；其次，它支持嵌入图片、音频、视频等多媒体内容，增强网页的表现力；第三，它具备平台无关性，编写的网页可以在任何操作系统和设备上通过 Web 浏览器访问；第四，它提供多种方式扩展功能，例如通过 JavaScript 增加交互性；第五，它支持多种语言和字符集，适用于创建多语言网页；最后，其语法相对简单，初学者可以快速掌握并上手使用。总的来说，HTML 在互联网的各个方面都起到了至关

重要的作用<sup>[3]</sup>。

## 2.2 CSS

JavaScript (JS) 是一种高级、动态的编程语言，广泛用于网页开发以增加交互性和动态功能。作为客户端脚本语言，JavaScript 运行在用户的浏览器中，通过操作 DOM（文档对象模型）来动态修改网页内容和样式。它的事件驱动特性允许捕获用户操作并作出响应，从而实现丰富的交互体验。除了网页开发，JavaScript 还能够用于服务器端编程（如 Node.js）和桌面应用开发。其丰富的生态系统，包括大量的第三方库和框架，如 React、Angular 和 Vue.js 等，极大地提升了开发效率和应用的复杂度。通过这些工具，开发者可以快速构建现代化的 Web 应用程序。JavaScript 与 HTML 和 CSS 共同构成了 Web 开发的基础，是现代互联网不可或缺的一部分<sup>[4]</sup>。

## 2.3 Java Script

JavaScript，通常缩写为 JS，是一种高级的、解释型的编程语言。作为一种基于对象和原型的动态脚本语言，它主要用于网页上实现客户端的脚本编程，以增强网页的交互性和功能性。JavaScript 具有简洁的语法和丰富的功能，可以用于表单验证、动态内容更新（如 AJAX）、动态样式更改和控制文档对象模型（DOM）。它还支持创建复杂的用户界面、数据处理和动画效果，甚至可以用于 Web 游戏开发。通过 JavaScript，开发者可以为用户提供丰富的浏览体验，并构建高度互动和响应灵敏的应用程序。JavaScript 的跨平台特性使其可以在各种浏览器上运行，无论用户使用的是哪种操作系统。随着 Web 技术的不断发展，JavaScript 已经成为 Web 开发的核心组成部分，拥有庞大的开发者社区和成熟的生态系统。不仅如此，JavaScript 还扩展到了服务器端，通过 Node.js 等技术实现全栈开发。它还在移动开发领域得到广泛应用，如 React Native 等框架。总而言之，JavaScript 是现代 Web 开发不可或缺的一部分，它的持续发展和创新推动着整个互联网技术的进步<sup>[5]</sup>。

## 2.4 Git Bash 代码管理

Git Bash 是一个在 Windows 操作系统上模拟 Unix/Linux 环境的命令行工具，旨在为开发者提供一个熟悉且强大的平台来使用 Git 进行代码管理。它不仅可

以简化 Git 操作，如提交、推送、拉取和分支管理等，还提供了许多其他功能，如远程仓库交互、自定义别名和颜色显示等。通过 Git Bash，开发者可以轻松地进行初始化和管理工作，追踪文件的变化历史，并在需要时回溯到之前的版本。它还支持团队协作开发，通过分支和合并操作，使得团队成员能够并行地开发和管理代码。内置的 SSH 客户端进一步简化了与远程仓库的交互，使用户能够安全地推送和拉取代码。此外，Git Bash 还支持自定义别名和颜色显示，提高了命令的可读性和工作效率。总而言之，Git Bash 将 Git 的强大功能与 Unix/Linux 风格的命令行环境相结合，为 Windows 用户提供了一个高效、灵活且易用的代码管理工具，使得开发者可以更加专注于代码本身，提高开发效率和代码质量<sup>[6]</sup>。

## 2.5 Visual Studio Code

Visual Studio Code（简称 VS Code）是由微软开发的一款免费、开源的跨平台代码编辑器，被广泛用于各种软件开发领域。VS Code 的主要特点之一是其高度可扩展性，用户可以根据自己的需求选择并安装各种插件来增强编辑器的功能，如语法高亮、代码补全、调试工具、版本控制集成等。这使得 VS Code 成为一个灵活且适用于多种编程语言的开发环境。VS Code 还内置了智能代码补全（IntelliSense）和错误检测功能，大大提高了编码的准确性和效率。它还支持多窗口和分屏编辑，方便开发者同时查看和编辑多个文件。此外，VS Code 还集成了终端窗口，开发者可以直接在编辑器中执行命令和调试代码，无需切换到其他工具。VS Code 的界面简洁直观，操作流畅，并且对于不同的操作系统都提供了相似的使用体验，这为开发者提供了一个一致的开发环境。总而言之，Visual Studio Code 通过其强大的功能、丰富的扩展生态系统和良好的用户体验，成为了许多开发者的首选编辑器。无论是初学者还是专业开发者，都可以借助 VS Code 的优势来提高开发效率、编写高质量的代码。它的活跃社区和持续的更新也保证了其与时俱进，不断满足开发者的需求<sup>[7]</sup>。

## 3. 增量式迭代开发模式(ADIT)

增量式迭代开发模式在复杂软件开发中的应用具有显著优势，尤其在本科毕业设计中，通过将整体开发过程分解为多个小的、可管理的增量，每个增量



都包含需求分析、设计、实现、测试和部署等完整的开发周期。这种方法强调灵活性和适应性，允许在每次迭代中根据实际情况灵活调整需求和设计，快速响应变化。频繁交付可用的产品增量可以及时获取用户反馈，确保产品不断优化，符合用户需求。通过持续集成与严格测试，每次迭代后保证了新功能能够无缝集成到现有系统中，维持系统的稳定性和一致性，有效管理开发过程中可能出现的风险。具体实施过程中，项目团队在第一迭代中确认基本功能需求并搭建基础框架，随后逐步实现用户管理、数据存储、核心业务功能和用户界面等模块。在最后几次迭代中，针对性能优化和安全需求进行设计和实现，确保最终产品的高质量和安全性。增量式迭代开发模式使问题可以在早期被发现和解决，减少了后期大规模重构的可能性，提高了开发效率和质量。在本项目中，采用这种模式不仅提高了开发效率，还确保了最终产品的高质量和用户满意度，证明了其在复杂软件开发中的有效性和优越性。通过这种模式，项目团队能够逐步交付高质量的软件产品，确保开发过程中的灵活性和适应性，从而达到预期的开发目标。在整个开发过程中，频繁的用户反馈和持续的测试保证了系统质量的不断提升，有效降低了开发风险，最终成功实现了高质量的软件作品。这一过程中，不仅培养了学生的团队协作和项目管理能力，还使他们深入理解了现代软件开发理念，为未来职业发展打下坚实基础。

## 4. “三段论”式的内容设计与实现

在本科毕业设计中，项目采用三段式开发模式，包括需求分析、代码实现和运行与测试。首先，通过详细的需求分析明确项目目标和功能范围；接着，按照需求文档进行系统设计和迭代编码，每个增量版本都经过严格的测试；最后，进行全面的运行与测试，通过单元测试、集成测试和用户验收测试确保系统稳定性和性能，持续优化以满足用户需求。

### 4.1 分析与设计

#### 4.1.1. 分析

本项目的 UI 设计遵循用户友好的“三段论”原则，旨在提供一个优秀的用户体验。首先，通过醒目的标题性信息，如 logo 或文字标题，迅速吸引用户的

注意力。这个部分非常重要，因为它是用户与界面首次接触的地方，能够在瞬间传达出项目的核心理念和特点。其次，核心内容区是整个 UI 设计的焦点，我们坚信“内容为王”的理念，致力于提供高质量的内容，以吸引和留住用户。这包括清晰而有吸引力的页面布局、易读的字体和高质量的图像。最后，底部附加信息区提供了对用户可能关心的细节和变化的展示，以满足用户对信息完整性的需求。此外，我们注重 UI 设计的一致性和可扩展性。在保持整体风格统一的同时，通过模块化的设计方法，可以灵活地添加或调整功能模块，以满足不断变化的业务需求。通过精心设计每个部分，本项目致力于打造一个既美观又实用的 UI 界面，让用户在使用过程中感到舒适和愉悦。我们相信这种 UI 设计方法将为用户提供出色的体验，并为未来的发展提供了可持续的基础。

总之，本项目在 UI 设计上追求简洁、清晰、一致，同时兼顾内容质量和细节展示。本项目相信，通过精心的设计和不断的优化，可以为用户提供一个优质的使用体验，帮助项目取得成功，如下图 4-1 第一次增量迭代用例图所示。

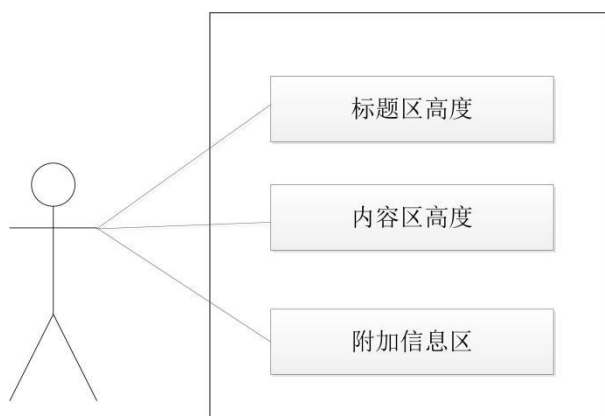
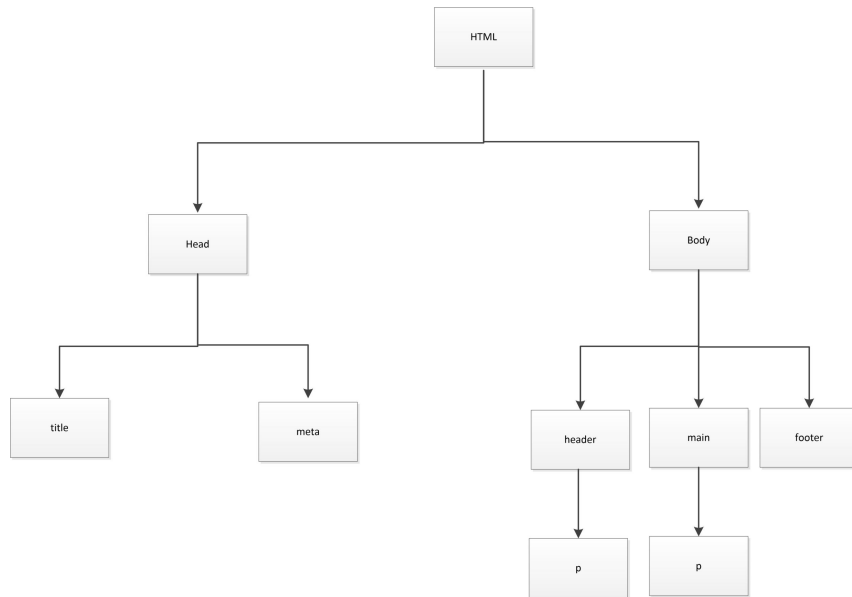


图 4-1 第一次增量迭代用例图

### 4.1.2. 设计

首先，使用 HTML5 来定义页面的布局框架。在这个框架中，需要添加 header、main 和 footer 这三个关键的布局元素。header 元素用于展示网站的标题，main 元素承载核心的艺术作品内容，而 footer 则用于展示版权信息和其他个人信息。随后，设计编写 CSS 对这些元素进行了细致的样式设计，以确保网页的视觉效果与如图 4-2 项目 Dom 树所示。



图表 4-2 第一次增量迭代 Dom 树

在 Dom 树里面可以清晰地了解项目结构。

## 4.2 代码实现

### 4.2.1 HTML 代码

```
<header>
  作品展示
</header>
<main>
  内容展示区
</main>
<footer>
  &copy; 彭锦鹏 江西科技师范大学 2024-2025
</footer>
<a href="1.2.html" target="_blank">1.2.html</a>
<a href="1.3.html" target="_blank">1.3.html</a>
<a href="1.4.html" target="_blank">1.4.html</a>
<a href="1.5.html" target="_blank">1.5.html</a>
<a href="1.6.html" target="_blank">1.6.html</a>
```

### 4.2.2 CSS 代码

```
* {
margin: 2px;
text-align: center;
align-content: center;
justify-content: center;
font-size: 30px;
font-family: 'KaiTi', '楷体', 'SimKai', 'STKaiti', serif;
```

```
}  
header {  
border: 4px solid black;  
height: 180px;  
}  
main {  
border: 4px solid black;  
height: 750px;  
}  
footer {  
border: 4px solid black;  
height: 100px;  
}  
a {  
display: inline-block;  
padding: 10px;  
color: black;  
box-shadow: 2px 2px 5px 0px rgba(0, 0, 0, 0.3);  
background-color: #4a6987;  
background-image: linear-gradient(62deg, #486b8e 0%, #5e4477 100%);  
border-radius: 10px;  
text-decoration: none;  
text-decoration: none;  
}
```

### 4.3 运行与测试

通过 gitbash 来提交自己的日志记录，如图 4-3，再通过直接向 GitHub 提交代码，实现互联网访问。

```
$ git log  
commit 7a71c75bd45c99e027eff4e4032728c702df0129 (HEAD -> master)  
Author: 20213625彭锦鹏@科师大 <2981435477@qq.com>  
Date: Fri Jun 14 16:53:53 2024 +0800
```

第一次代码提交,我们完成了软件的设计概要,完成了三部分。**header**部分放了软件的标题,**main**部分放了软件的内容, **footer**放了软件的动态反馈。

图 4-3 提交日志

在浏览器中访问部署到 GitHub 上面的网址，可以直接访问增量迭代第一版，效果图如下图 4-4 运行效果图。



图 4-4 运行效果图

手机端可以扫描下图 4-5 阶段 1 增量迭代二维码，访问“三段论”式的内容设计与实现文件。



图 4-5 阶段 1 增量迭代二维码

## 5. 响应式设计与适应窄屏实现

### 5.1 分析与设计

#### 5.1.1. 分析

响应式设计是一种现代化的网页设计方法，旨在使网站能够在各种设备上提供一致和优质的用户体验。通过使用流体网格布局、弹性图片和媒体查询等技术，响应式设计使得网页能够根据用户访问设备的特性自动调整布局和内容呈现方式，从而确保在桌面电脑、平板电脑和智能手机等各种不同屏幕尺寸和

分辨率下均能够良好显示和操作。这种方法不仅简化了网站维护和更新的复杂性，还提高了用户的访问便利性和满意度，因为用户无需手动调整页面大小或水平滚动来查看内容。响应式设计的普及和应用，推动了网页设计领域向更加灵活和用户友好的方向发展，成为现代网络体验设计的重要组成部分<sup>[8]</sup>。

本次更新迭代代码要实现在不同用户的不同设备上正确运行，给用户最佳体验，因此要完成响应式设计并且适应窄屏的自动调节。如下图 5-1 第二次迭代项目用例图所示。

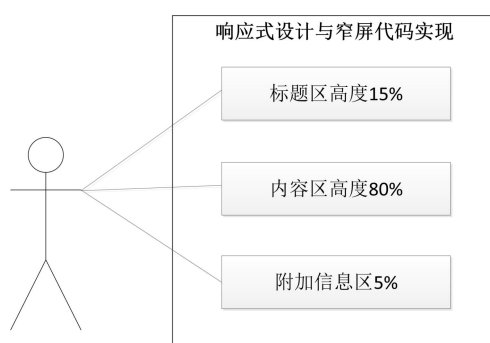


图 5-1 第二次增量迭代项目用例图

### 5.1.2 设计

首先，使用 HTML5 来定义页面的布局框架。在前面的第一次迭代中分别在 header 中添加了 p 标签。如图 4-2 项目 Dom 树所示。

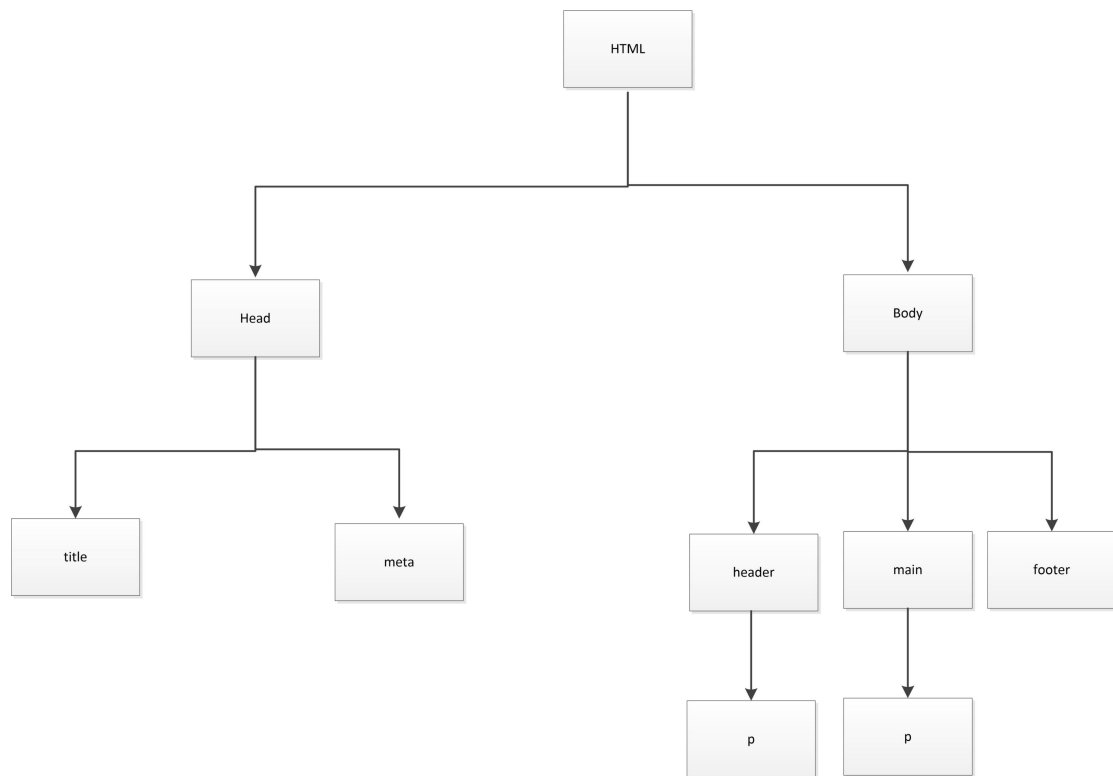


图 4-2 项目 Dom 树

## 5.2 代码实现

### 5.2.1 HTML 代码

```
<header>
<p id="head">
作品展示
</p>
</header>
<main id='main'>
<p class="Content">
内容展示区
</p>
</main>
<footer>
&copy 彭锦鹏 江西科技师范大学 2024--2025
</footer>
```

### 5.2.2 CSS 代码

```
*{
margin: 2px;
text-align: center;
align-content: center;
justify-content: center;
font-family: 'KaiTi', '楷体', 'SimKai', 'STKaiti', serif;
```

```

box-sizing: border-box;
-webkit-user-select: none;
/* Safari */
-moz-user-select: none;
/* Firefox */
-ms-user-select: none;
/* Internet Explorer/Edge */
user-select: none;
/* 标准语法 */
}

header {
background-color: #54a57a;
background-image: linear-gradient(45deg, #85FFBD 0%, #FFFB7D 100%);
border: 2px solid black;
height: 15%;
font-size: 1.66em;
}

#head {
font-size: 1.3em;
}

main {
background-color: rgb(247, 247, 247);
border: 2px solid black;
height: 80%;
font-size: 1.2em;
display: flex;
}

.Content {
font-size: 1.1em;
width: 75%;
background-color: #d1869f;
background-image: linear-gradient(0deg, #FFDEE9 0%, #203c3b 100%);
box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.4);
}

footer {
background-color: #FBAB7E;
background-image: linear-gradient(62deg, #FBAB7E 0%, #F7CE68 100%);
border: 2px solid black;
height: 5%;

```



```
}
```

### 5.2.3 JavaScript 代码

```
var UI = {};
UI.appWidth = window.innerWidth > 600 ? 600 : window.innerWidth;
UI.appHeight = window.innerHeight;
const LETTERS = 22;
const baseFont = UI.appWidth / LETTERS;
//通过更改 body 对象的字体大小，这个属性能够遗传其子子孙孙
document.body.style.fontSize = baseFont + "px";
//通过把 body 对象的宽度和高度设置为设备/屏幕的宽度和高度，实现全屏。
//通过 CSS 对子对象百分比（纵向）的配合，从而实现响应式设计的目标。
document.body.style.width = UI.appWidth + "px";
document.body.style.height = UI.appHeight - 1 * baseFont + "px";
```

## 5.3 运行与测试

通过 gitbash 来提交自己的日志记录，如图 5-2，再通过直接向 GitHub 提交代码，实现互联网访问。

```
Author: 20213625彭锦鹏@科师大 <2981435477@qq.com>
Date: Fri Jun 14 18:00:16 2024 +0800

第二次增量迭代--本次提交了'移动互联时代的响应式设计(窄屏)'代码,本次代码运行了应用程序的响应式设计:1.声明了一个全局UI对象,用来存放需要使用的属性,2.为四个区域设置了占整个用户界面的比例(以高度为准)。3.用Javascript获取了应用屏幕的宽度和高度,并依据需要来进行计算基础字体的大小。4.在不同的区域根据内容的重要程度设定了该区域字体的相对大小。5.改变了颜色和对比色,对组区域的背景图做了优化外观的设计。6.总之,初步为用户的应用程序做了响应式设计并部署了代码。
```

图 5-2 提交日志

这一小节展示在不同的设备上面运行项目的效果图，这是增量迭代的第二版，分别有 iPhone SE、iPhone 12 Pro 在这两种设备上面运行本项目，实际效果与预期效果别无二致。效果图如下图 5-3 iPhone SE 与 iPhone 12 Pro 的效果对比图所示。

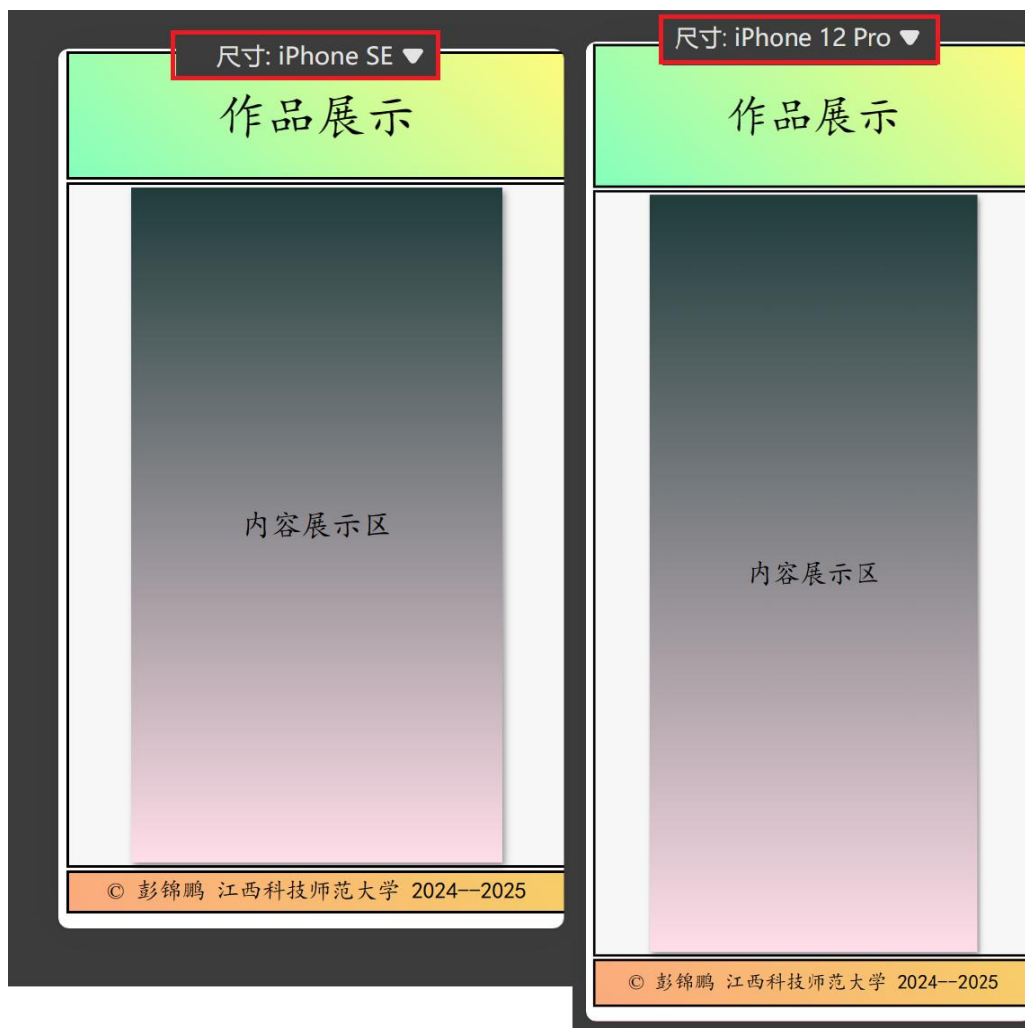


图 5-3 iPhone SE 与 iPhone 12 Pro 的效果对比图

分析图 5-1 iPhone SE 与 iPhone 12 Pro 的效果对比图可知，代码运行正常，实际效果与预期效果一致。代码可以根据用户的设备屏幕大小，来自动调整最佳页面效果，展示给用户。

手机端可以扫描下图 5-4 阶段 2 增量迭代二维码，访问响应式设计与适应窄屏实现的文件。

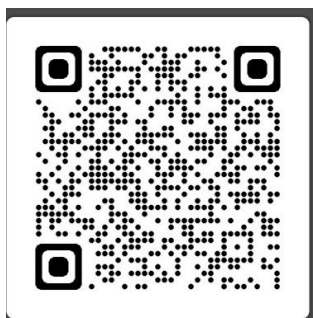


图 5-4 阶段 2 增量迭代二维码

## 6. 个性化化交互 UI 设计——鼠标模型

### 6.1 分析与设计

#### 6.1.1 分析

用户期望能够通过网页浏览作者的作品内容，了解作者的创作风格和作品特点。用户希望能够在不同设备上都能够流畅地浏览作品，体验良好的用户界面。同时，网页需要提供作品展示功能，包括展示作品的标题、图片等内容。同时，需要实现全屏展示和响应式设计，确保用户在不同设备上都能够获取良好的浏览体验。此外，需要实现鼠标控制 UI 功能，让用户可以通过鼠标操作进行交互。网页设计需要使用 HTML、CSS 和 JavaScript 等前端技术，实现页面布局、样式设置和交互功能。需要确保代码的可读性和可维护性，以便今后进行扩展和更新。同时，需要考虑代码的兼容性，确保在不同浏览器上都能够正常运行。对于用户的个人信息和数据需要进行保护，确保用户在浏览网页时不会受到安全威胁。需要采取相应的安全措施，如数据加密和防火墙等，保障用户信息的安全<sup>[9]</sup>。

如下图 6-1 第三次增量迭代用例图所示。

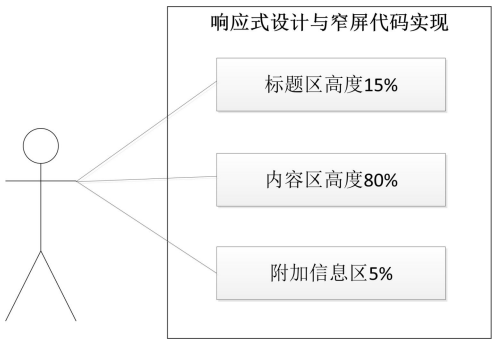


图 6-1 第三次增量迭代用例图

#### 6.1.2 设计

在设计用户界面时，我采用了一种直观的方法来展示鼠标坐标信息，通过在主容器<main>的子元素<p>中动态更新这些数据。为此，我定义了一个 mouse 对象来捕获和保存鼠标状态和位置信息，包括鼠标按下状态、当前 X 坐标和水平移动增量。接着，我为鼠标的 mousedown、mousemove 和 mouseout 事件注册了监听器，以便在鼠标操作的不同阶段进行响应。利用<p>元素的 textContent

属性，我能够将捕获到的鼠标信息实时反映在页面上，使用户在进行鼠标操作时能够立即看到其坐标变化。这种设计不仅增强了用户的交互体验，而且通过清晰的 DOM 树结构，提供了一种模块化、易于管理和扩展的解决方案。

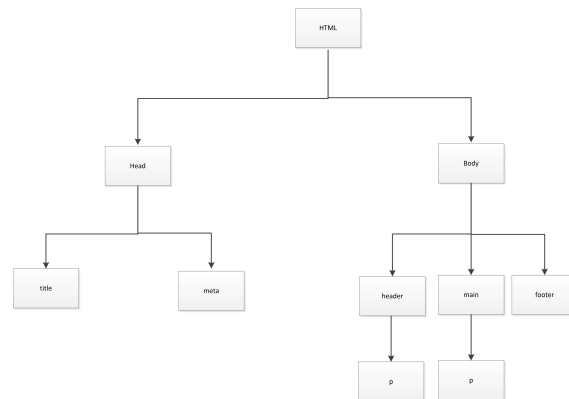


图 6-2 项目 Dom 树所示

## 6.2 代码实现

### 6.2.1 HTML 代码

```

header>
  <p id="head">
    作品展示
  </p>
</header>
<main id="main">
  <p class="Content">
    内容展示区
  </p>
</main>
<footer>
  &copy; 彭锦鹏 江西科技师范大学 2024--2025
</footer>
  
```

### 6.2.2 CSS 代码

```

* {
margin: 2px;
text-align: center;
align-content: center;
justify-content: center;
font-family: 'KaiTi', '楷体', 'SimKai', 'STKaiti', serif;
box-sizing: border-box;
-webkit-user-select: none;
/* Safari */
  
```

```

-moz-user-select: none;
/* Firefox */
-ms-user-select: none;
/* Internet Explorer/Edge */
user-select: none;
/* 标准语法 */
}

header {
border: 2px solid black;
height: 15%;
background-color: aqua;
font-size: 1.66em;
box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.2);
}

main {
background-color: rgb(247, 247, 247);
border: 2px solid black;
height: 80%;
font-size: 1.2em;
display: flex;
box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.2);
}

.Content {
font-size: 1.1em;
width: 75%;
background-color: #FFDEE9;
background-image: linear-gradient(0deg, #FFDEE9 0%, #B5FFFC 100%);
box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.4);
}

footer {
border: 2px solid black;
height: 5%;
background-color: chocolate;
box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.2);
}

```

### 6.2.3 JavaScript 代码

```

var UI = {};
UI.appWidth = window.innerWidth > 600 ? 600 : window.innerWidth;
UI.appHeight = window.innerHeight;

```

```

const LETTERS = 22;
const baseFont = UI.appWidth / LETTERS;

//通过更改 body 对象的字体大小，这个属性能够遗传其子子孙孙
document.body.style.fontSize = baseFont + "px";
//通过把 body 对象的宽度和高度设置为设备/屏幕的宽度和高度，实现全屏。
//通过 CSS 对子对象百分比（纵向）的配合，从而实现响应式设计的目标。
document.body.style.width = UI.appWidth + "px";
document.body.style.height = UI.appHeight - 1 * baseFont + "px";

//尝试对鼠标设计 UI 控制
var mouse = {};
mouse.isDown = false;
mouse.x = 0;
mouse.deltaX = 0;
$(".Content").addEventListener("mousedown", function(ev) {
let x = ev.pageX;
let y = ev.pageY;

console.log("鼠标按下了，坐标为: " + "(" + x + "," + y + ")");
$(".Content").textContent = "鼠标按下了，坐标为: " + "(" + x + "," + y + ")";
});
$(".Content").addEventListener("mousemove", function(ev) {
let x = ev.pageX;
let y = ev.pageY;

console.log("鼠标正在移动，坐标为: " + "(" + x + "," + y + ")");
$(".Content").textContent = "鼠标正在移动，坐标为: " + "(" + x + "," + y + ")";
});
$(".Content").addEventListener("mouseout", function(ev) {
//console.log(ev);
$(".Content").textContent = "鼠标已经离开";
});
$("body").addEventListener("keypress", function(ev) {
let k = ev.key;
let c = ev.keyCode;
$(".keyboard").textContent = "您的按键 : " + k + " , " + "字符编码 : " + c;
});

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 $}
```

## 6.3 运行与测试

通过 gitbash 来提交自己的日志记录，如图 6-3，再通过直接向 GitHub 提交代码，实现互联网访问。

```
Author: 20213625彭锦鹏@科师大 <2981435477@qq.com>
Date: Sun Jun 16 09:48:55 2024 +0800
```

第三次增量迭代实现了一个作品展示网页的基本结构和交互功能。通过设置全屏展示和响应式设计，确保用户在不同设备上具有良好的浏览体验。通过监听鼠标事件和键盘事件，实现了对用户操作的响应。同时，通过自定义的\$函数，简化了页面元素的获取过程。整体功能较为简单，但为用户提供了基本的作品展示和交互体验。

图 6-3 提交日志图

在浏览器中访问部署到 GitHub 上面的网址，可以直接访问增量迭代第二版，效果图如下图 6-4 正常电脑运行效果图所示。



图 6-4 正常电脑运行效果图

这一小节展示在不同的设备上面运行项目的效果图，分别有 iPhone 12 Pro、

iPhone XR 在这两种设备上面运行本项目，实际效果与预期效果别无二致。效果图如下图 6-5 iPhone 12 Pro 与 iPhone XR 的效果对比图所示。

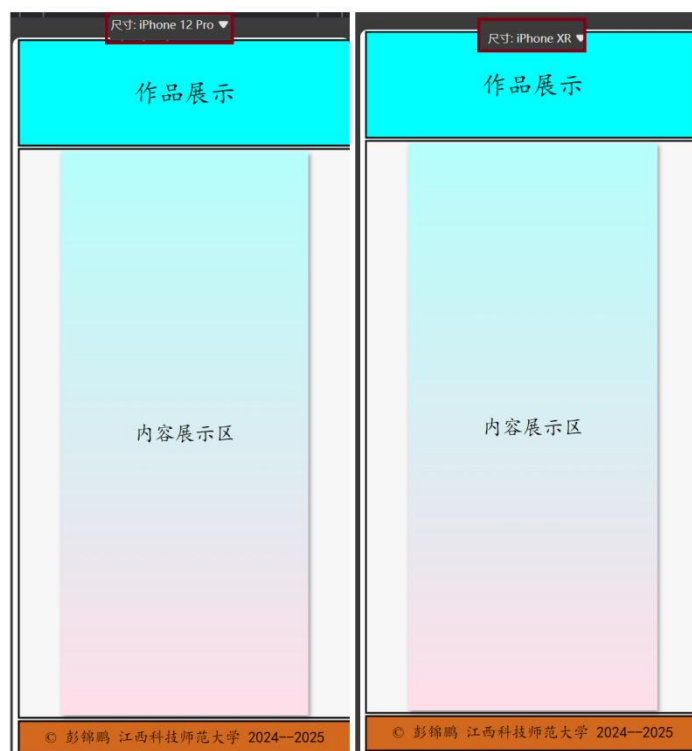


图 6-5 iPhone 12 Pro 与 iPhone XR 的效果对比图

在图 6-5 中可知，系统已经满足了前面需求分析的要求，第三次增量迭代在第二次增量迭代的基础上，增加了鼠标点击和移动控制等功能，大大的增加了用户体验感，可以通过控制鼠标来为下一次增量迭代做出铺垫。

手机端可以扫描下图 6-6 阶段 3 增量迭代二维码，访问适用移动互联时代的响应式设计文件。



图 6-6 阶段 3 增量迭代二维码

## 7. 通用 UI 设计为触屏和鼠标统一建模



## 7.1 分析与设计

### 7.1.1 分析

为了提高 Web 应用开发的效率和软件质量，统一建模被应用于确立一致性标准和最佳实践。这种方法旨在增强代码的可维护性、可读性和可扩展性，同时促进团队协作和代码复用。为了简化调试和问题解决，统一建模还致力于创建易于理解和维护的代码结构。如下图 7-1 项目第 5 次增量用例图所示。

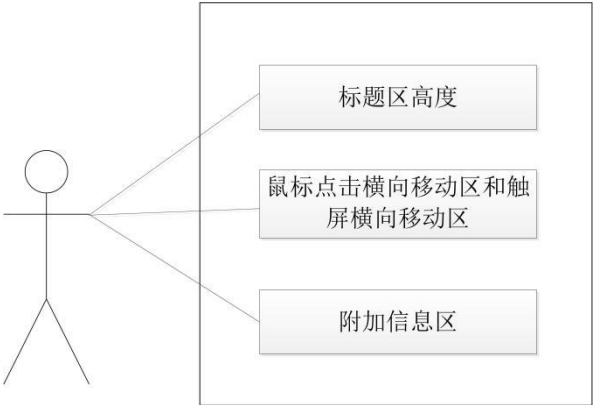


图 7-1 第 5 次增量项目用例图

### 7.1.2 设计

第 5 次迭代更新代码主要实现了一套基于触摸和鼠标事件的交互逻辑，允许用户通过拖动或滑动来控制页面上的元素。此外，代码中还包含了一个自定义的\$函数，用于简化 DOM 元素的选择过程。如下图 7-2 项目 Dom 树所示。

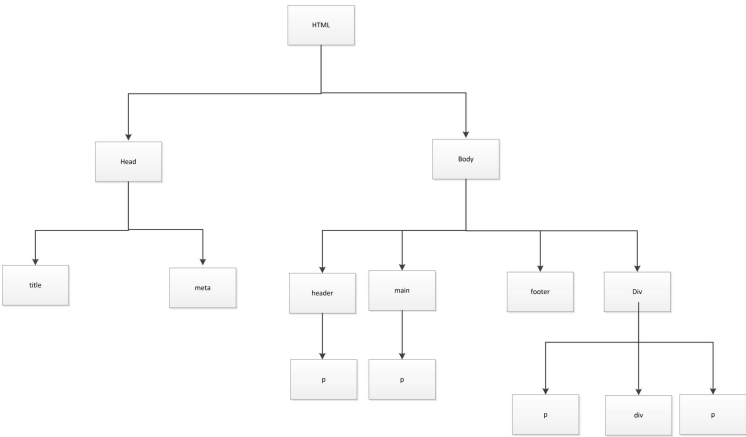


图 7-2 项目 Dom 树

## 7.2 代码实现

### 7.2.1 HTML 代码

```
<header>
```

```
<p id="head">
作品展示
</p>
</header>
<main id="main">
<div class="Content">
内容展示区
</div>
</main>

<footer>
&copy; 彭锦鹏 江西科技师范大学 2024--2025
</footer>
<div id="aid">
<p id="userResponseArea">用户键盘响应区</p>
<div id="keyboard">
<p id="displaytxt">
&nbsp;
</p>
</div>
<p id="displayCode">
&nbsp;
</p>
</div>
```

### 7.2.2 CSS 代码

```
* {
margin: 2px;
text-align: center;
align-content: center;
justify-content: center;
font-family: 'KaiTi', '楷体', 'SimKai', 'STKaiti', serif;
box-sizing: border-box;
-webkit-user-select: none;
/* Safari */
-moz-user-select: none;
/* Firefox */
-ms-user-select: none;
/* Internet Explorer/Edge */
user-select: none;
/* 标准语法 */
}

header {
```

```
background-color: #85FFBD;
background-image: linear-gradient(45deg, #85FFBD 0%, #FFFB7D 100%);
border: 2px solid black;
height: 15%;
font-size: 1.66em;
box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.2);
}

main {
background-color: rgb(247, 247, 247);
border: 2px solid black;
height: 80%;
font-size: 1.2em;
display: flex;
box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.2);
}

.Content {
font-size: 1.1em;
position: relative;
width: 75%;
background-color: #8a4b60;
background-image: linear-gradient(0deg, #764b59 0%, #72c0bd 100%);
box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.4);
}

footer {
background-color: #FBAB7E;
background-image: linear-gradient(62deg, #FBAB7E 0%, #F7CE68 100%);
border: 2px solid black;
height: 5%;
box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.2);
}

#aid {
top: 0.3em;
left: 650px;
position: absolute;
align-content: normal;
border: 2px solid black;
box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.2);
}

#userResponseArea {
margin: 0.8em;
font-size: 1.66em;
```

```

}
#keyboard {
  border: 3px solid black;
  margin-left: 10%;
  width: 80%;
  height: 60%;
  box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.4);
}
#displayCode {
  border: 3px solid black;
  margin-left: 10%;
  width: 80%;
  position: absolute;
  bottom: 0.3em;
  box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.4);
}
</style>

```

### 7.2.3 JavaScript 代码

```

var UI = {};
UI.appWidth = window.innerWidth > 600 ? 600 : window.innerWidth;
UI.appHeight = window.innerHeight;
const LETTERS = 22;
const baseFont = UI.appWidth / LETTERS;
//通过更改 body 对象的字体大小，这个属性能够遗传其子子孙孙
document.body.style.fontSize = baseFont + "px";
//通过把 body 对象的宽度和高度设置为设备/屏幕的宽度和高度，实现全屏。
//通过 CSS 对子对象百分比（纵向）的配合，从而实现响应式设计的目标。
document.body.style.width = UI.appWidth + "px";
document.body.style.height = UI.appHeight - 1 * baseFont + "px";
UI.appHeight = window.innerHeight;
if (window.innerWidth < 1000) {
  $("aid").style.display = 'none';
}
$("aid").style.width = window.innerWidth - UI.appWidth - baseFont * 3 + 'px';
$("aid").style.height = UI.appHeight - baseFont * 1 + '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;
$(".Content").addEventListener("mousedown", function(ev) {

```

```

mouse.isDown = true;
mouse.x = ev.pageX;
mouse.y = ev.pageY;
console.log("mouseDown at x: " + "(" + mouse.x + "," + mouse.y + ")");
$(".Content").textContent = "鼠标按下, 坐标: " + "(" + mouse.x + "," + mo ";
});

$(".Content").addEventListener("mouseup", function(ev) {
mouse.isDown = false;
$(".Content").style.left = 0 + 'px';
$(".Content").textContent = "鼠标松开!";
if (Math.abs(mouse.deltaX) > 100) {
$(".Content").textContent += " 这是有效拖动! ";
} else {
$(".Content").textContent += " 本次算无效拖动! ";
}
});

$(".Content").addEventListener("mouseout", function(ev) {
ev.preventDefault();
mouse.isDown = false;
$(".Content").textContent = "鼠标里开!";
if (Math.abs(mouse.deltaX) > 100) {
$(".Content").textContent += " 这次是有效拖动! ";
} else {
$(".Content").textContent += " 本次算无效拖动! ";
}
});

$(".Content").addEventListener("mousemove", function(ev) {
ev.preventDefault();
if (mouse.isDown) {
console.log("mouse isDown and moving");
mouse.deltaX = parseInt(ev.pageX - mouse.x);
$(".Content").textContent = "正在拖动鼠标, 距离: " + mouse.deltaX + "px 。 ";
$(".Content").style.left = mouse.deltaX + 'px';
}
});
// 触屏事件处理
var touch = {
startX: 0,
startY: 0,
deltaX: 0
};

```

```

$(".Content").addEventListener("touchstart", function(ev) {
  var touchEvent = ev.touches[0];
  touch.startX = touchEvent.pageX;
  touch.startY = touchEvent.pageY;
  $(".Content").textContent = "触屏开始, 坐标: " + "(" + parseInt(touch.startX, 10) +
  parseInt(touch.startY, 10) + ")";
});

$(".Content").addEventListener("touchmove", function(ev) {
  ev.preventDefault();
  var touchEvent = ev.touches[0];
  touch.deltaX = parseInt(touchEvent.pageX - touch.startX);
  $(".Content").textContent = "正在触屏滑动, 距离: " + touch.deltaX + "px 。";
  $(".Content").style.left = touch.deltaX + 'px';
});

$(".Content").addEventListener("touchend", function(ev) {
  $(".Content").textContent = "触屏结束!";
  if (Math.abs(touch.deltaX) > 100) {
    $(".Content").textContent += ", 这是有效滑动! ";
  } else {
    $(".Content").textContent += " 本次算无效滑动! ";
  }
});

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 $

```

### 7.3 运行与测试

在本节内容中，我将上一次迭代实现的实时捕捉并显示触摸点的坐标信息与触屏类响应的实时信息显示在屏幕上的功能后端代码整合成一套代码，也就是通用代码的实现，减少 JavaScript 的运行压力。下面分别展示 iPhone XR 设备与电脑响应设备运行效果与差异，页面效果如下图 7-3 所示。

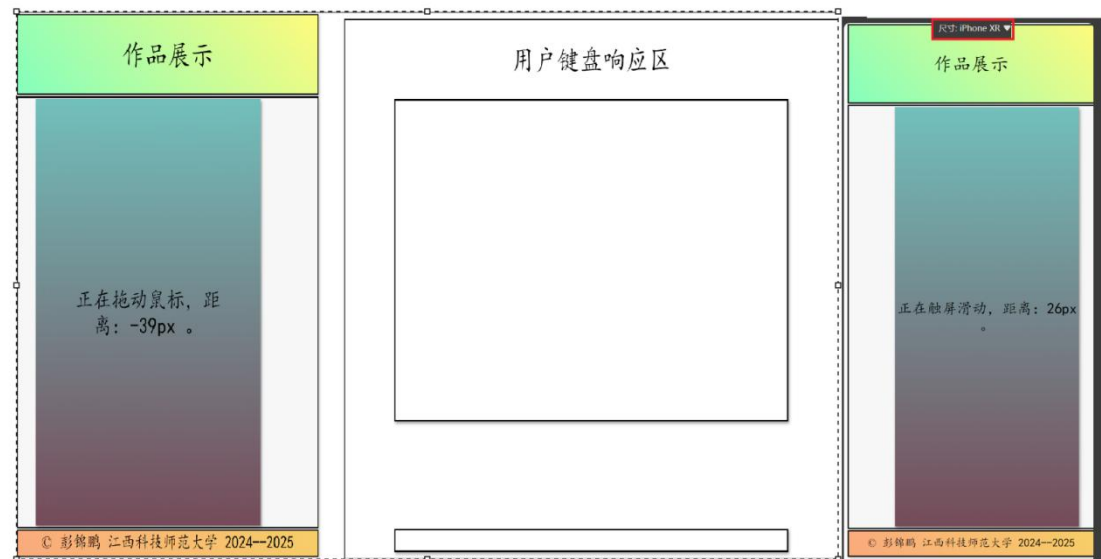


图 7-3 页面效果

分析图 7-3 页面效果可知，代码运行正常，实际效果与预期效果一致。代码可以根据模拟触屏调节内容区的位置，运行效果良好。

手机端可以扫描下图 7-4 阶段 5 增量迭代二维码，访问 UI 拖动模拟触屏后的操作模式页面。



图 7-4 阶段 5 增量迭代二维码

在本节中，我开发了一个触屏模拟系统，该系统通过设计的算法和用户界面技术，确保了网页在各种设备上都能提供卓越的触摸交互体验。通过集成复杂的事件处理程序和动态坐标跟踪机制，系统能够准确识别和响应用户的每一

个触摸动作，无论是轻触、滑动还是长按。此外，系统还特别优化了在不同屏幕尺寸和分辨率下的显示效果和交互逻辑，确保用户在浏览网页或完成特定任务时，都能获得一致、流畅且直观的触觉反馈。这一创新方法大大提升了用户与数字内容之间的互动质量，使得触屏操作更加自然和直观。

通过 gitbash 来提交自己的日志记录，如图 7-5，再通过直接向 GitHub 提交代码，实现互联网访问。

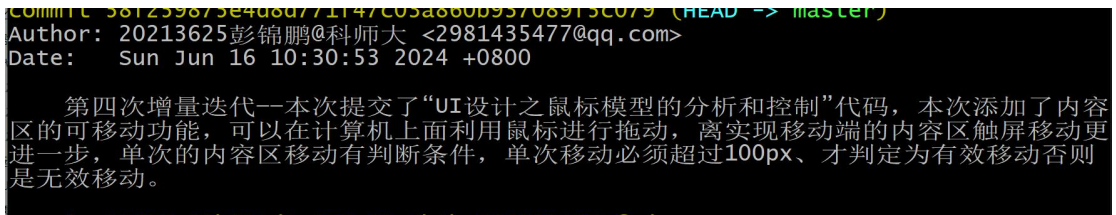


图 7-5 提交代码图

## 8. 个性化 UI 监控键盘实现

### 8.1 分析与设计

#### 8.1.1 分析

键盘作为计算机及多种电子设备的核心输入工具，扮演着至关重要的角色。它不仅为用户提供了一个直观、高效的文本和指令输入手段，而且通过功能键和快捷键的辅助，极大提升了操作的灵活性和工作效率<sup>[10]</sup>。

为了深入掌握键盘的工作原理及其在用户界面中的实现方式，我专门添加了一个交互式的用户键盘响应区域，用以探究和演示键盘输入如何影响页面信息的展示。该设计包括了详尽的用户界面布局和响应机制，如图 8.1 描绘，旨在通过直观的展示和动态反馈，增强用户对键盘事件处理流程的认识。通过这一实践，用户不仅能够观察到按键行为如何被系统捕获和解析，还能够看到这些行为如何触发特定的页面反应，从而在提升用户体验的同时，也加深了对键盘交互背后技术的理解。



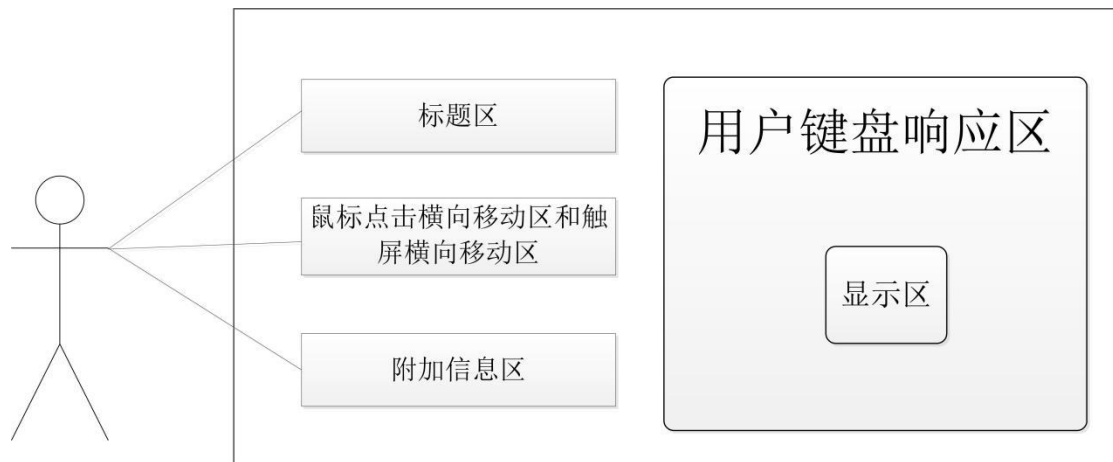
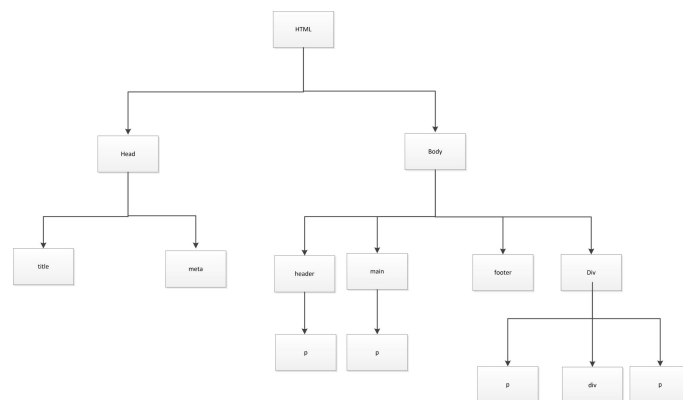


图 8.1 键盘响应应用例图

### 8.1.2 设计

这段代码通过监听键盘的按下和释放事件，添加了一个实时响应用户输入的 Web 界面。它能够捕捉并显示用户按下的键及其编码，同时在页面上的特定元素中动态展示输入内容。系统特别设计了对 Enter 键的换行和 Backspace 键的删除功能，同时确保只有有效的字符（包括字母、数字和选定的标点符号）被添加到显示区域。此外，自定义的\$函数简化了 DOM 元素的选择过程，增强了代码的健壮性。实现过程不仅提升了用户交互体验，还通过即时反馈增强了界面的直观性和互动性。如下图 8-2 项目 Dom 树所示。



图如下图 8-2 项目 Dom 树所示

## 8.2 代码实现

### 8.2.1 HTML 代码

```
<header>
  <p id="head">
    作品展示
  </p>
</header>
<main id="main">
```

```
<div class="Content">
  内容展示区
</div>
</main>
<footer>
  &copy; 彭锦鹏 江西科技师范大学 2024--2025
</footer>
<div id="aid">
  <p id="userResponseArea">用户键盘响应区</p>
  <div id="keyboard">
    <p id="displaytxt">
      &nbsp;
    </p>
  </div>
  <p id="displayCode">
    &nbsp;
  </p>
</div>
```

### 8.2.2 CSS 代码

```
* {
  margin: 2px;
  text-align: center;
  align-content: center;
  justify-content: center;
  font-family: 'KaiTi', '楷体', 'SimKai', 'STKaiti', serif;
  box-sizing: border-box;
  -webkit-user-select: none;
  /* Safari */
  -moz-user-select: none;
  /* Firefox */
  -ms-user-select: none;
  /* Internet Explorer/Edge */
  user-select: none;
  /* 标准语法 */
}

header {
  background-color: yellowgreen;
  border: 2px solid black;
  height: 15%;
  font-size: 1.66em;
  box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.2);
}
```

```

main {
background-color: rgb(247, 247, 247);
border: 2px solid black;
height: 80%;
font-size: 1.2em;
display: flex;
box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.2);
}

.Content {
font-size: 1.1em;
position: relative;
width: 75%;
background-color: #FFDEE9;
background-image: linear-gradient(0deg, #295675 0%, #B5FFFC 100%);
box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.4);
}

footer {
border: 2px solid black;
background-color: coral;
height: 5%;
box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.2);
}

#aid {
top: 0.3em;
left: 650px;
position: absolute;
align-content: normal;
border: 2px solid black;
box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.2);
}

#userResponseArea {
margin: 0.8em;
font-size: 1.66em;
}

#keyboard {
background-color: #74EBD5;
background-image: linear-gradient(90deg, #4dc4ae 0%, #9FACE6 100%);
border: 3px solid black;
margin-left: 10%;
width: 80%;
}

```

```
height: 60%;
box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.4);
}

#displayCode {
border: 3px solid black;
margin-left: 10%;
width: 80%;
position: absolute;
bottom: 0.3em;
box-shadow: 2px 3px 5px 0px rgba(0, 0, 0, 0.4);
}
```

### 8.2.3 JavaScript 代码

```
// 创建一个名为 UI 的对象，用于存储系统的宽度和高度信息
var UI = {};
// 记录系统窗口的宽度和高度，并限制宽度不超过 600px
UI.appWidth = window.innerWidth > 600 ? 600 : window.innerWidth;
UI.appHeight = window.innerHeight;
// 计算默认字体大小
const Letters = 22; // 字母数量
const baseFont = UI.appWidth / Letters; // 基准字体大小
// 设置页面的字体大小为默认字体大小
document.body.style.fontSize = baseFont + 'px';
// 通过动态 CSS 设置页面全屏显示
document.body.style.width = UI.appWidth + 'px';
document.body.style.height = UI.appHeight - 1 * baseFont + "px";
if (window.innerWidth < 1000) {
  $("#aid").style.display = 'none';
}
$("#aid").style.width = window.innerWidth - UI.appWidth - baseFont * 3 + 'px';
$("#aid").style.height = UI.appHeight - 1 * baseFont + '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;
$(".Content").addEventListener("mousedown", function(ev) {
  mouse.isDown = true;
  mouse.x = ev.pageX;
  mouse.y = ev.pageY;
  console.log("mouseDown at x: " + "(" + mouse.x + "," + mouse.y + ")");
  $(".Content").textContent = "鼠标按下, 坐标: " + "(" + mouse.x + "," + mouse.y + ")";
```

```

});
$(".Content").addEventListener("mouseup", function(ev) {
    mouse.isDown = false;
    $(".Content").textContent = "鼠标松开!";
    $(".Content").style.left = 0 + 'px';
    if (Math.abs(mouse.deltaX) > 100) {
        $(".Content").textContent += "，这是有效拖动! ";
    } else {
        $(".Content").textContent += " 本次算无效拖动! ";
    }
});
$(".Content").addEventListener("mouseout", function(ev) {
    ev.preventDefault();
    mouse.isDown = false;
    $(".Content").textContent = "鼠标离开!";
    if (Math.abs(mouse.deltaX) > 100) {
        $(".Content").textContent += " 这次是有效拖动! ";
    } else {
        $(".Content").textContent += " 本次算无效拖动! ";
    }
});
$(".Content").addEventListener("mousemove", function(ev) {
    ev.preventDefault();
    if (mouse.isDown) {
        console.log("mouse isDown and moving");
        mouse.deltaX = parseInt(ev.pageX - mouse.x);
        $(".Content").textContent = "正在拖动鼠标，距离: " + mouse.deltaX + "px 。";
        $(".Content").style.left = mouse.deltaX + 'px';
    }
});

$("body").addEventListener("keypress", function(ev) {
    $(".displaytxt").textContent += ev.key;
});
$("body").addEventListener("keydown", function(ev) {
    let k = ev.key;
    let c = ev.keyCode;
    $(".displayCode").textContent = "您已按下键 : " + k + " , " + "字符编码 : " + c;
});
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;
}
}
}
```

### 8.3 运行与测试

在本章节，我们深入探索了键盘交互的内在机制，开发了一个创新的键盘响应展示平台。该平台位于网页的侧面展示区域 **main**，通过一个精心设计的 `<div>` 标签，动态捕捉并可视化用户的键盘敲击动作。这种直观的展示不仅加深了用户对键盘输入过程的理解，而且通过实时的页面更新，提供了一种新颖的反馈形式，极大地丰富了人机交互的深度和广度。用户现在可以直观地观察到每个按键如何触发页面元素的变化，从而在提升操作透明度的同时，也增加了交互的趣味性。如下图 8-3 页面效果所示。

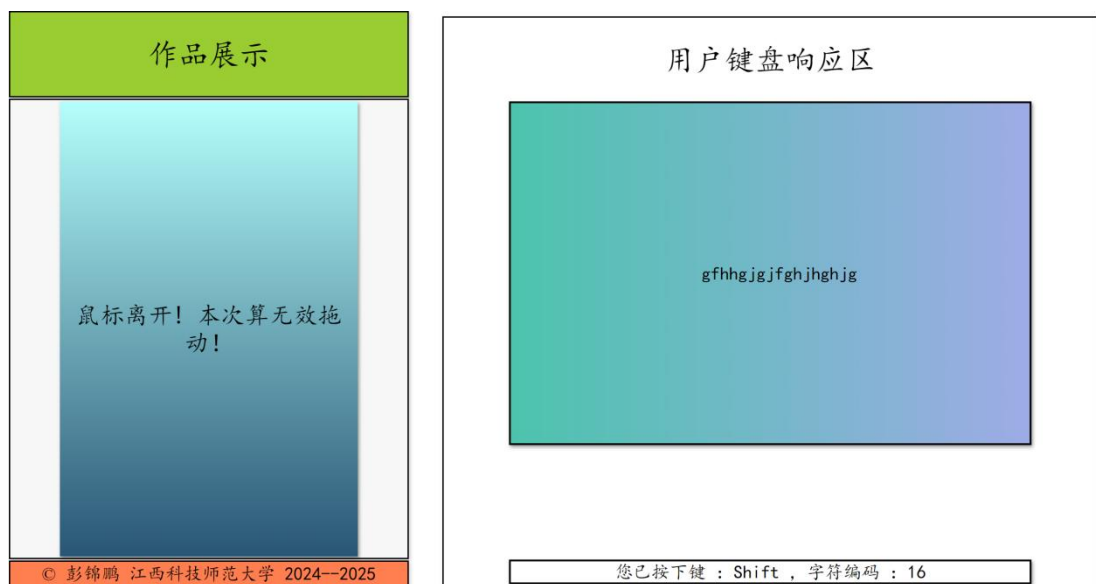


图 8-3 页面效果

分析图 8-3 页面效果可知，代码运行正常，实际效果与预期效果一致。代码可以根据模拟触屏调节内容区的位置，运行效果良好。

手机端可以扫描下图 8-4 阶段 6 增量迭代二维码，访问 UI 拖动模拟触屏后的操作模式页面。



图 8-4 阶段 6 增量迭代二维码

## 8.4 代码提交与版本控制

通过 gitbash 来提交自己的日志记录，如图 7-5，再通过直接向 GitHub 提交代码，实现互联网访问。



图 8-5 阶段 6 提交日志图

## 9. 用 Git Bash 工具管理项目的代码仓库和 Http 服务器

### 9.1 跨世纪的经典 Bash 工具

Bash (Bourne Again Shell) 是一种功能强大且广泛应用的 Unix Shell，最初于 1989 年发布，并迅速成为大多数 Linux 发行版的默认 Shell。它不仅是一个命令解释器，也提供了一个完整的编程环境，支持丰富的特性和工具，使用户能够高效地进行系统管理和开发任务。Bash 的管道机制允许将一个命令的输出直接作为下一个命令的输入，通过这种方式，可以构建复杂的数据处理流水线。同时，Bash 支持输入、输出和错误重定向，使得文件读写和错误信息处理

更加灵活。

**Bash** 脚本编写功能使得用户能够自动化许多任务，如定时备份、系统监控和批量处理数据。它支持变量、条件语句、循环和函数等编程结构，非常适合用来编写各种自动化脚本。此外，**Bash** 还包含一系列经典的文本处理工具，如 **grep**、**sed** 和 **awk**，这些工具各有其独特的用途：**grep** 用于搜索和模式匹配，**sed** 用于流编辑和替换，**awk** 则是一个功能强大的文本处理语言。这些工具经常与管道结合使用，形成高效的数据处理流水线。

在文件处理方面，**Bash** 提供了诸如 **cp**、**mv**、**rm** 和 **find** 等工具，用于复制、移动、删除和查找文件，极大地方便了文件系统的管理。**Bash** 还支持作业控制，使用户可以在前台和后台运行命令，并通过 **jobs**、**fg** 和 **bg** 等命令管理作业。此外，**Bash** 的定时任务调度功能，通过 **cron** 和 **at** 命令，允许用户设置定时执行的任务，为系统管理和维护提供了极大便利。

环境变量在 **Bash** 中也起着重要作用，它们存储了配置信息，如路径变量 **PATH** 决定了可执行文件的查找路径。用户可以通过编辑配置文件（如 **.bashrc**、**.bash\_profile**）来自定义 **Shell** 环境，加载别名、函数和启动脚本。**Bash** 的命令历史记录功能使得用户可以方便地访问之前输入的命令，而自动补全功能（通过 **Tab** 键）进一步提高了操作的效率和准确性。

总之，经过数十年的发展和优化，**Bash** 已成为现代计算机使用和管理中不可或缺的一部分。它强大的功能和灵活的特性，不仅提升了用户的工作效率，还为复杂任务的自动化和简化提供了坚实的支持

## 9.2 通过 Git Hub 平台实现本项目的全球域名。

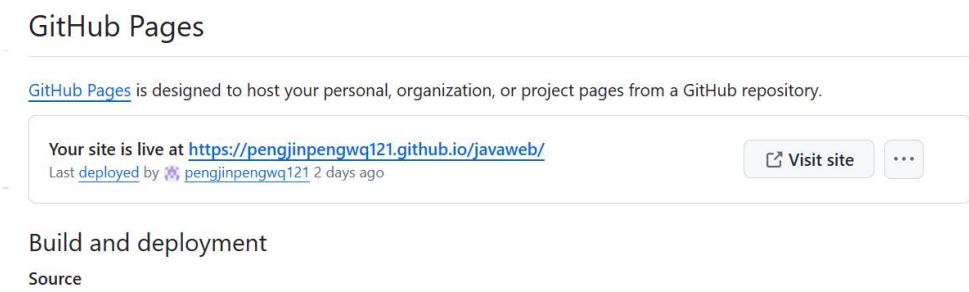


图 9.1 本项目的全球域名

如上图 9.1 所示，为本项目在 Git Hub 平台上的全球域名，通过图 9-2 可以在 PC 端的微软 Edge 浏览器中输入该域名进行访问。



从此，我们无论在本地做了任何多次代码修改，也无论提交了多少次，上传远程时都会把这些代码和修改的历史记录全部上传git hub平台，而远程上传命令则可简化为一条：`git push`，极大地方便了本Web应用的互联网发布。

远程代码上传后，项目可以说免费便捷地实现了在互联网的部署，用户可以通过域名或二维码打开，本次使用PC的微软Edge浏览器打开，如下图 9.3 所示：



图 9-2 最终作品

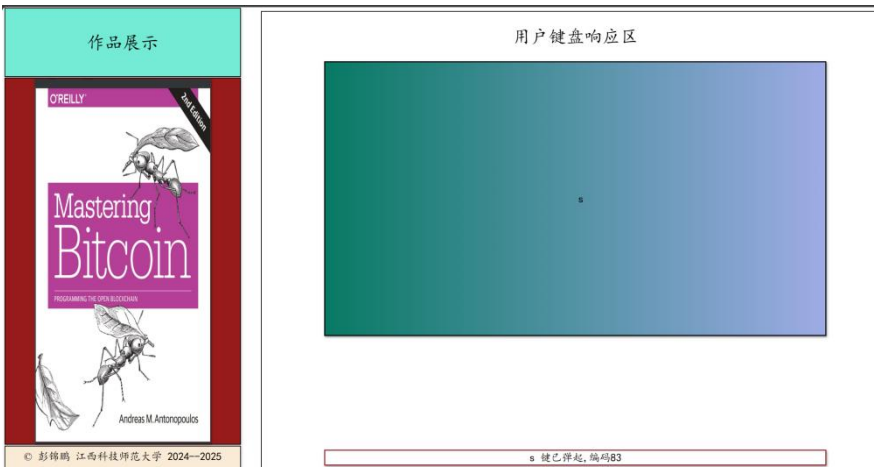


图 9.3 项目在互联网的部署

## 参考文献

- [1] 段易含, 刘桂芝. 我国拔尖创新人才培养的政策协同及其优化[J]. 湖南师范大学教育科学学报, 2024.
- [2] 闵克香, 冯扬. 赋能未来: 上海图书馆人工智能素养教育实践与思考[J]. 晋图学刊, 2024.
- [3] 杨宏宇, 张建伟, 胡泽, 等. Web 应用攻击检测方法综述[J]. 中国民航大学学报, 2024,42(02):13-22.
- [4] 杨宏宇, 张建伟, 胡泽, 等. Web 应用攻击检测方法综述[J]. 中国民航大学学报, 2024,42(02):13-22.
- [5] 印志鸿, 白云璐. 新编大学计算机信息技术教程[M]. 南京大学出版社, 2022.
- [6] 吴迪, 陈晓民, 肖群力. 外资云服务在中国落地实务[M]. 世界图书出版公司, 2022.
- [7] 徐小龙, 李洋, 林皓伟, 等. 云数据中心智能管理[M]. 电子工业出版社, 2021.
- [8] 汪瑞霞. 第二届“生态文明引领下的乡村振兴”国际研讨会暨设计助力乡村建设青年论坛优秀论文集[M]. 江苏凤凰美术出版社, 2022.
- [9] 蔡睿, 葛军, 孙哲, 等. AI 预训练大模型发展综述[J]. 小型微型计算机系统, 2024.
- [10] 高文静, 卢永鑫. 视觉传达技艺在现代产品设计中的应用[J]. 鞋类工艺与设计, 2023,3(23):21-23.