

成绩:

江西科技师范大学

毕业论文（设计）

题目（中文）：基于 web 客户端技术的个性化 UI 设计与实现

（外文）：Personalized UI design and implementation

院（系）：元宇宙产业学院

专 业：计算机科学与技术

学生姓名：周龙卿

学 号：20213605

指导教师：李建宏

2024 年 6 月 12 日

目 录

1. 引言	4
1.1. 项目概述	4
1.2. 研学计划	5
1.3. 研究方法	5
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
2.3. 文献综述	8
3. 内容设计概要	9
3.1. 分析和设计	9
3.2. 项目的实现和编程	9
3.3. 项目的运行和测试	11
3.4. 项目的代码提交和版本管理	11
4. 移动互联时代的 UI 开发初步——窄屏终端的响应式设计	12
4.1. 分析和设计	12
4.2. 项目的实现和编程	13
4.3. 项目的运行和测试	14
4.4. 项目的代码提交和版本管理	15
5. 应用响应式设计技术开发可适配窄屏和宽屏的 UI	17
5.1. 分析和设计	17
5.2. 项目的实现和编程	17
5.3. 项目的运行和测试	20
5.4. 项目的代码提交和版本管理	20
6. 个性化 UI 设计——鼠标模型	22
6.1. 分析和设计	22
6.2. 项目的实现和编程	22
6.3. 项目的运行和测试	25
6.4. 项目的代码提交和版本管理	25

7. 对触屏和鼠标的通用交互操作的设计开发	27
7.1. 分析和设计	27
7.2. 项目的实现和编程	27
7.3. 项目的运行和测试	30
7.4. 项目的代码提交和版本管理	30
8. UI 的个性化键盘交互控制的设计与开发	32
8.1. 分析和设计	32
8.2. 项目的实现和编程	32
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	35
9. 谈谈本项目的高质量代码	37
9.1. 清晰的错误结构和注释	37
9.2. 代码规范和标准	37
9.3. 代码的可测试性	38
9.4. 代码性能优化	38
9.5. 代码版本控制	38
10. 用 gitBash 工具管理项目的代码仓库和 http 服务器	39
10.1. 跨世纪的经典 Bash 工具	39
10.2. 利用 gitBash 工具对本项目进行管理	40
10.2.1. 项目文件的基本搭建	40
10.2.2. 项目代码的提交与上传	40
10.2.3. 项目日志的反馈查询	41
10.3. 通过 GitHub 平台实现本项目的全球域名	41
10.3.1. 创建空的远程代码仓库	错误！未定义书签。
10.3.2. 设置本地仓库与远程代码仓库的连接并推送	41
参考文献	43

Web 技术的个性化 UI 设计

摘要：个性化和响应式的用户界面设计在现代数字化和移动互联的背景下变得必不可少。本论文研究了基于 Web 客户端技术的个性化 UI 设计与开发，根据用户的偏好和行为习惯，定制独特的界面元素和交互方式。根据用户的偏好和行为习惯，定制独特的界面元素和交互方式。本文首先概述了 Web 平台的核心技术，包括 HTML5、CSS3 和 JavaScript。文章详细描述了响应式设计如何通过媒体查询和弹性布局技术，确保 Web 应用能够自动适应从窄屏智能手机到宽屏桌面显示器的各种设备。针对个性化 UI，论文分析了如何利用用户数据和行为模式进行界面元素的定制，满足不同用户的操作习惯和视觉偏好，尤其是多种输入方式如鼠标、触控和键盘的统一设计，确保 Web 应用在不同交互设备上都能提供流畅一致的用户体验。此外，项目采用了增量式迭代开发模型，利用 GitBash 工具进行版本控制和项目管理，通过 GitHub 平台实现代码的全球托管与协作。最终，本文通过实际案例展示了个性化 UI 设计的实施效果，并进行了测试和验证，结果表明基于 Web 技术的个性化 UI 设计不仅提升了用户体验，还能适应多样化的设备需求，为未来 Web 应用的开发提供了宝贵的参考。

关键词：Web 技术；HTML；CSS；JavaScript；个性化；不同用户端

Personalized UI design for Web technology

Abstract: In the context of modern digitalization and mobile connectivity, personalized and responsive user interface (UI) design has become critical. This thesis studies the design and development of personalized UI based on Web client technology, aiming to create a high quality user experience that is consistent across devices. This article begins with an overview of the core technologies of the Web platform, including HTML5, CSS3. It then details how responsive design, through media queries and resilient layout techniques, ensures that Web applications automatically adapt to a wide range of devices, from narrow-screen smartphones to widescreen desktop displays. For personalized UI, this paper analyzes how to customize interface elements by using user data and behavior patterns to meet different users' operating habits and visual preferences, especially the unified design of multiple input methods such as mouse, touch and keyboard, so as to ensure that Web applications can provide smooth and consistent user experience on different interactive devices. In addition, the project adopts an incremental iterative development model, uses GitBash tools for version control and project management, and realizes global hosting and collaboration of code through GitHub platform. Finally, this paper shows the implementation effect of personalized UI design through actual cases, and carries out testing and verification. The results show that personalized UI design based on Web technology not only improves user experience, but also ADAPTS to diversified device requirements, providing a valuable reference for the development of future Web applications.

Key words: Web technology; HTML; CSS; JavaScript; Personalization; Different clients

1. 引言

1.1. 项目概述

在当今信息化、数字化的时代,Web 技术作为连接用户与互联网的重要桥梁,其界面设计(UI 设计)的个性化已成为提升用户体验、增强用户粘性的关键因素。个性化 UI 设计不仅涉及美观与功能的融合,更涉及对用户需求的深入理解和精准把握。本论文旨在探讨 Web 技术中个性化 UI 设计的理念、方法与实践,以为相关领域的研究和实践提供有价值的参考。

本项目的主要目标包括:

1. 个性化交互: 根据用户的操作习惯和偏好,动态调整 UI 布局和交互方式,提供个性化的用户体验。
2. 多种输入支持: 开发支持鼠标、键盘和触屏的通用交互模式,确保用户能够在各种设备上进行操作。
3. 用户中心: 个性化 UI 设计始终围绕用户的需求和体验展开,确保界面元素和交互方式符合用户的心理和行为模式。
4. 独特性和创新性: 个性化 UI 设计追求独特性和创新性,通过独特的界面设计和交互方式吸引用户的眼球并提升用户体验。
5. 响应式设计: 设计和实现能够适应不同屏幕尺寸(如手机、平板和桌面设备)的用户界面,确保在各种设备上的一致性和可用性。
6. 功能性和友好性: 个性化 UI 设计不仅注重美观和独特性,还要确保界面具备强大的功能性和良好的用户友好性,使用户能够轻松完成操作任务。

通过本论文的研究,我们期望能够加深对个性化 UI 设计的理解和认识,为 Web 设计领域的研究和实践提供新的思路和方法。同时,我们也期望通过本论文的推广和应用,推动 Web 技术的个性化发展,为用户带来更加优质、个性化的互联网体验。

1.2. 研学计划

为确保项目的顺利进行并实现预期目标，收集和阅读相关文献，了解 Web 客户端技术和个性化 UI 设计的最新发展。分析已有案例，明确项目的功能需求和用户体验目标。制定系统架构设计，确定整体框架和模块划分。选择合适的前端框架、CSS 预处理器及版本控制工具（如 Git）。开发系统原型，初步实现主要界面和基本功能。收集初步用户反馈，调整和优化原型设计。实现项目的核心功能模块，如用户管理、数据处理、个性化推荐等。进行模块集成测试，确保功能的正确性和稳定性。设计和实现响应式 UI，适配不同屏幕尺寸。优化系统性能，提高响应速度和用户体验^[3]。

1.3. 研究方法

首先，通过相关的 Web 客户端技术在个性化 UI 设计和实现的文献的研究，确定用户对移动互联时代的个性化 UI 设计的需求和期望，研究分析项目的可行性，并通过使用软件工程中常用的开发方法中的经典模型设计：迭代增量式开发模型，它将整个软件开发过程分为多个小步骤（迭代），每个迭代都会产生一个可执行的部分（增量），并且在后续的迭代中逐步完善和扩展这个部分。基于定性和定量研究所得数据，进行统计分析和数据挖掘，根据其中所得的相应数据，通过使用迭代增量式开发模型来逐步实现移动互联时代中个性化 UI 设计，最后总结归纳 Web 客户端技术中个性化 UI 设计和实现的关键问题和未来发展方向，探讨可能的发展趋势和创新方向，为相关研究和实践提供启示和指导。

2. 技术总结和文献综述

2.1. Web 平台和客户端技术概述

Web 之父 Tim Berners-Lee 在发明 Web 的基本技术架构以后，就成立了 W3C

组织，该组织在 2010 年后推出的 HTML5 国际标准，结合欧洲 ECMA 组织维护的 ECMAScript 国际标准，几乎完美缔造了全球开发者实现开发平台统一的理想，直到今天，科学家与 Web 行业也还一直在致力于完善这个伟大而光荣的理想[1]。学习 Web 标准和 Web 技术，学习编写 Web 程序和应用有关工具，最终架构一套高质量代码的跨平台运行的应用，是我的毕设项目应用的技术路线。Web 应用的程序设计体系由三大语言有机组成：HTML, CSS, JavaScript。这三大语言的组合也体现了人类社会化大生产分工的智慧，可以看作用三套相对独立体系实现了对一个信息系统的描述和控制，可以总结为：HTML 用来描述结构（Structure）、CSS 用来描述外表（presentation）、Javascript 用来描述行为（Behavior）[3]；这也可以用经典的 MVC 设计模式来理解 Web 平台架构的三大基石，Model 可以理解为 HTML 标记语言建模，View 可以理解为用 CSS 语言来实现外观，Controller 则可理解为用 JavaScript 结合前面二个层次，实现了在微观和功能层面的代码控制。

在现代 Web 开发中，基于 Web 客户端技术的个性化 UI 设计已成为一个热门话题。以下是几个关键技术的总结：前端框架：诸如 React、Vue.js 和 Angular 等前端框架提供了强大的 UI 组件和状态管理功能，极大地简化了前端开发的复杂性，并促进了组件化和模块化的开发方式。响应式设计：响应式设计是一种使网站能够根据不同的设备（如电脑、平板电脑和手机）和屏幕尺寸自动调整其布局和样式的技术。通过媒体查询和流式布局等技术，实现了网站在不同设备上的良好显示效果。增量式迭代开发：增量式迭代开发是一种将项目分解成多个增量，逐步开发、测试和集成的开发模式。每个增量都包含了完整的开发周期，有助于提高项目的灵活性、可控性和可维护性。版本控制和代码管理：采用 Git 等版本控制工具进行代码管理，能够有效地协调多人开发、追踪代码修改历史、管理分支和解决冲突，为团队协作提供了良好的基础。

2.2. 项目的增量式迭代开发模式

本项目作为一个本科专业学生毕业设计的软件作品，与单一用途的程序相比较为复杂，本项目所涉及的手写代码量远超过简单一二个数量级以上，从分析问题的到初步尝试写代码也不是能在几天内能落实的，可以说本项目是一个系统工

程，因此需要从软件工程的管理视角来看待和规范项目的编写过程。

而本项目考虑选择的软件工程开发过程管理模式有两种经典模型：瀑布模型（The waterfall model）和增量式迭代模型(The incremental model)。而任何开发模式则都必须同样经历四个阶段：分析（Analysis）、设计（Design）、实施（Implementation）、测试（test）。

瀑布模型需要专业团队完美的配合，从分析、设计到实施，最后到测试，任何阶段的开始必须基于上一阶段的完美结束。而这对于我们大多数普通开发者是不太现实的，作为小微开发者由于身兼数职，其实无法一次就能完美完成任何阶段的工作，比如在实施过程中，开发者会发现前面的设计存在问题，则必须在下一次迭代项目时改良设计^[5]。在当今开源的软件开发环境中，开发者在软件的开发中总是在不断地优化设计、重构代码，持续改进程序的功能和代码质量。因此在本项目的开发中，也采用了增量模型的开发模式^[6]。本项目中我一共做了六次项目的开发迭代，如图 2.1 所示。

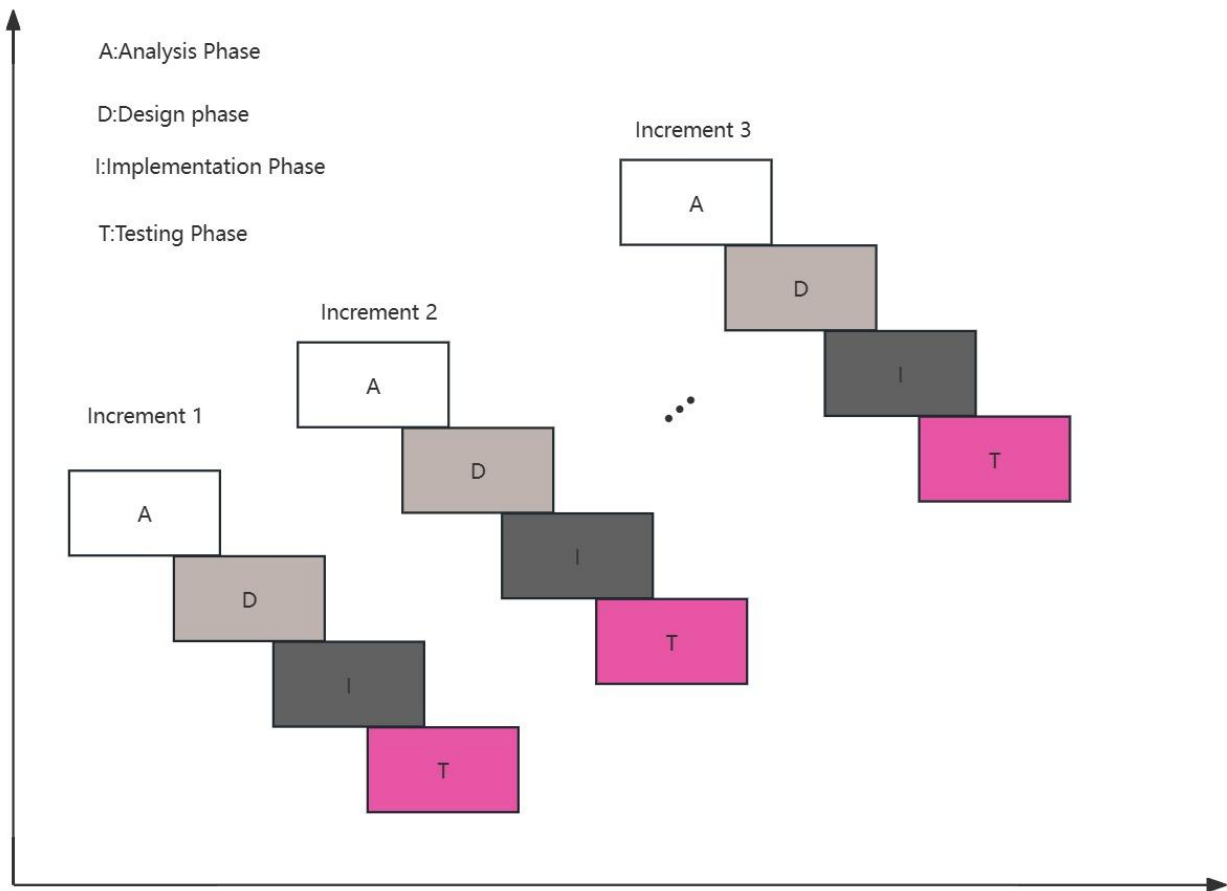


图 2.1 项目的开发迭代图

2.3. 文献综述

在个性化 UI 设计的领域，近年来 Web 开发技术的进步显著改变了用户界面的呈现和交互方式。随着 HTML5 和 CSS3 的广泛应用，Web 页面的设计变得更加灵活和功能丰富。例如，刘华星和杨庚（2011）指出，HTML5 的语义化标签和多媒体支持极大地提升了 Web 内容的表达能力和用户体验。黄永慧和陈程凯（2013）则探讨了 HTML5 在移动应用开发中的优势，强调了其在响应式设计中的关键作用。

JavaScript 及其框架的发展同样对 UI 设计产生了深远影响。现代前端框架如 React、Vue.js 和 Angular，不仅提高了开发效率，还通过组件化的方式增强了 UI 的可重用性和维护性（岳昆等，2004）。这些框架支持单页面应用（SPA）的开发，使得 Web 应用可以提供类似本地应用的流畅体验。

响应式设计作为 Web 开发中的重要理念，解决了不同设备之间的显示适配问题。张树明（2013）在其研究中阐述了响应式设计如何通过灵活的布局和媒体查询，实现 Web 页面在各种屏幕尺寸下的良好表现。这种设计方式在移动互联网时代尤为重要，能够显著提升用户的跨设备体验。

此外，个性化 UI 设计越来越重视用户的个性化需求和交互方式。张友生和李雄（2006）探讨了软件开发模型的演变，强调了增量式开发在快速迭代和用户反馈中的重要性。这与现代 UI 设计中对动态适应用户需求和实时调整界面的要求相一致。

总体而言，随着 Web 技术的持续发展，个性化 UI 设计正在不断进步，从早期的静态页面到如今的动态、响应式和个性化界面，Web 应用已经能够在多种设备和用户情境下提供一致且优质的用户体验。

3. 内容设计概要

3.1. 分析和设计

这一步是项目的初次开发，本项目最初使用人们习惯的“三段论”式简洁方式开展内容设计，首先用一个标题性信息展示 logo 或文字标题，吸引用户的注意力，迅速表达主题；然后展现主要区域，也就是内容区，“内容为王”是项目必须坚守的理念，也是整个 UI 应用的重点；最后则是足部的附加信息，用来显示一些用户可能关心的细节变化。如图 3.1 所示：

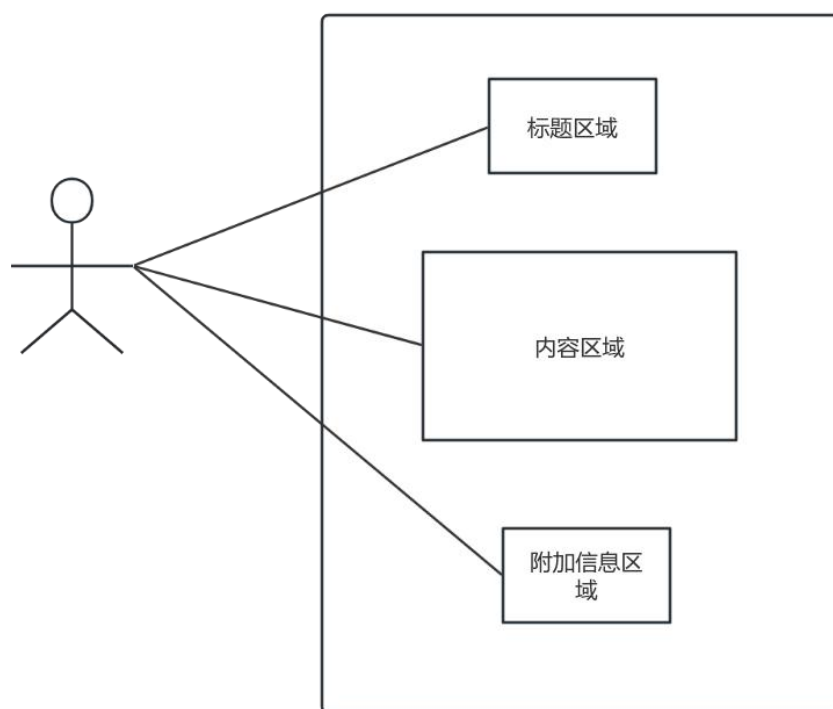


图 3.1 用例图

3.2. 项目的实现和编程

一、HTML 代码编写如下：

```
<header>
```

```
    《 我的毕设题目 》  
</header>  
  
<main>  
    我的主题内容： ‘读好书、练思维、勤编程’ @masterLijh 计算思维系列  
课程  
  
</main>  
  
<footer>  
    Copyright XXX 江西科技师范大学 2024-2025  
  
</footer>
```

二、CSS 代码编写如下：

```
*{  
margin: 10px;  
text-align: center;  
font-size: 30px ;  
}  
  
header{  
border: 2px solid blue;  
height: 200px;  
}  
  
main{  
border: 2px solid blue;  
height: 400px;  
}  
  
footer{  
border: 2px solid blue;  
height: 100px;  
}
```

```
a{
display: inline-block ;
padding:10px ;
color: white;
background-color: blue;
text-decoration: none ;
}
```

3.3. 项目的运行和测试

项目的运行和测试至少要通过二类终端，本文此处仅给出 PC 端用 Chrome 浏览器打开项目的结果，如图 3.2 所示。由于本项目的阶段性文件已经上传 github 网站，移动端用户可以通过扫描图的二维码，运行测试本项目的第一次开发的阶段性效果。如图 3.3 所示。



图 3.2 PC 端运行效果

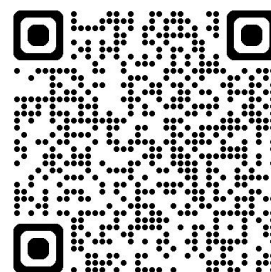


图 3.3 移动端二维码

3.4. 项目的代码提交和版本管理

本项目的文件通过 gitBash 工具管理，作为项目的第一次迭代，在代码提交和版本管理环节，我们的目标是建立项目的基本文件结构，还有设置好代码仓库的基本信息：如开发者的名字和电子邮件。

进入 gitBash 命令行后，按次序输入以下命令：

```
$ cd /  
$ mkdir webUI  
$ cd webUI  
$ git init  
$ git config user.name 江科师大周龙卿  
$ git config user.email 1652494674@qq.com  
$ touch index.html myCss.css
```

编写好 index.html 和 myCss.css 的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add index.html myCss.css  
$ git commit -m 项目第一版：“三段论”式的内容设计概要开发
```

成功提交代码后，gitbash 的反馈如下图 3.4 所示。

```
Administrator@WIN-SND7LHJPSHV MINGW64 /d/git—远程管理文件/github/WebUI (master)  
$ git commit -m '项目的第一版：'  
[master (root-commit) e9444c6] 项目的第一版：  
2 files changed, 150 insertions(+)  
create mode 100644 index.html  
create mode 100644 myCSS.css
```

图 3.4 gitbash 代码提交反馈

4. 移动互联时代的 UI 开发初步——窄屏终端的响应式设计

4.1. 分析和设计

随着移动设备的普及和网络技术的飞速发展，用户通过不同尺寸和类型的终端访问 Web 应用的需求日益增加。窄屏终端（如智能手机和平板电脑）在移动互联时代扮演着至关重要的角色。响应式设计成为适应这一趋势的重要技术手段，通过适配不同屏幕尺寸的 UI 设计，确保在各种设备上的一致性和优质用户体验^[7]。

4.2. 项目的实现和编程

一、HTML 代码编写如下：

```
<header>

<p id="book">
我的毕设题目
</p>
</header>

<nav>

<button>导航一</button>
<button>导航二</button>
<button>导航三</button>
</nav>

<main id = 'main'>
软件内容区域
</main>

<footer>

<p id="statusInfo">
周龙卿@ 江西科技师范大学 2025
</p>
</footer>
```

二、CSS 代码编写如下：

```
*{

margin: 10px;
text-align: center;
}

header{

border: 2px solid blue;
height: 15%;
```

```
font-size: 1.66em;
}
main{
border: 2px solid blue;
height: 70%;
font-size: 1.2em;
}
nav{
border: 2px solid blue;
height: 10%;
}
nav button{
font-size: 1.1em;
}
footer{
border: 2px solid blue;
height: 5%;
}
```

4.3. 项目的运行和测试

确保项目在不同设备和浏览器上的兼容性和性能是响应式设计的关键。在不同的浏览器（如 Chrome、Firefox、Safari、Edge）上测试 UI，确保一致的显示效果。在实际的移动设备上测试，确保 UI 在各种屏幕尺寸和分辨率上的表现。使用开发者工具（如 Chrome DevTools）中的设备模拟功能进行快速验证。使用工具（如 Lighthouse 或 PageSpeed Insights）分析页面的性能指标，如加载时间、交互延迟和渲染速度。实施代码分割和异步加载，提升应用的响应速度。通过用户测试收集实际用户的反馈，了解 UI 在不同设备上的使用体验。根据反馈进行调整和改进，提升用户满意度。其中，PC 端口如图 4.1 所示，移动端访问如图

4.2 所示。



图 4.1 移动端项目的 PC 端访问



图 4.2 移动端访问

4.4. 项目的代码提交和版本管理

编写好的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add 1.2.html
```

成功提交代码后，gitbash 的反馈如下图 4.3 所示：

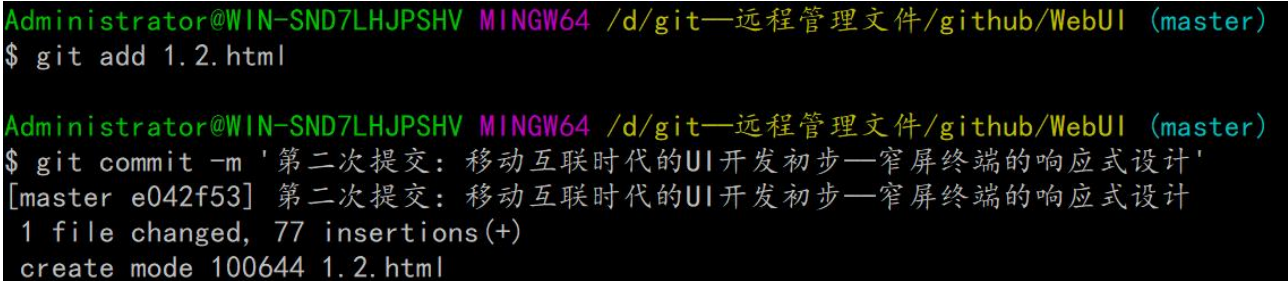


图 4.3 gitbash 提交反馈

gitbash 反馈代码的仓库日志如下所示：


```
$ git log
commit e042f534996fdf254d69dbf679139b5e018fd9f9 (HEAD -> master)
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 09:53:59 2024 +0800
```

第二次提交：移动互联时代的UI开发初步—窄屏终端的响应式设计

```
commit e9444c697cd8fd404c691f78abf5ce57608a4931
Author: youzelian <1431850082@qq.com>
Date:   Thu Jun 6 17:26:11 2024 +0800
```

项目的第一版：

图 4.4 gitbash 的仓库日志反馈

5. 应用响应式设计技术开发可适配窄屏和宽屏的 UI

5.1. 分析和设计

在当今这个复杂的设备上，创建能够适应从窄到宽、从智能手机到宽屏显示器等不同设备的用户界面(UI)至关重要。响应式设计通过一套适应性强的设计和技术，确保 Web 应用能够在不同设备上提供一致优化的用户体验。本章将把响应式设计在 UI 开发中的应用，包括从分析和设计，到实现、测试和版本管理的工作流程。

5.2. 项目的实现和编程

一、HTML 代码编写如下：

```
<header>
<p id="book">
  《我的毕设题目》
</p>
</header>

<nav>
<button>导航 1</button>
<button>导航 2</button>
<button>导航 3</button>
</nav>

<main id="main">
<div id="bookface">
  书的封面图
</div>
</main>

<footer>
```

```
</footer>
```

```
<div id="aid">
```

```
<p>用户键盘响应区</p>
```

```
<p id="keyboard"></p>
```

```
</div>
```

二、CSS 代码编写如下：

```
*{  
text-align: center;  
box-sizing: border-box ;  
}  
header,main,div#bookface,nav,footer{  
margin:1em;  
}  
header{  
border: 2px solid blue;  
height: 15%;  
font-size: 1.66em;  
  
}  
main{  
border: 2px solid blue;  
height: 70%;  
font-size: 1.2em;  
  
}  
nav{
```

```
border: 2px solid blue;
height: 10%;
}
nav button{
font-size: 1.1em;
}
footer{
border: 2px solid blue;
height: 5%;
}
body{
position:relative ;
}
#aid{
position: absolute;
border: 3px solid blue;
top: 0.5em;
left: 600px;
}
#bookface{
width: 80%;
height: 80%;
border: 1px solid red;
background-color: blanchedalmond;
margin:auto;
}
```

5.3. 项目的运行和测试

在实现阶段，我们将设计策略转化为具体的 HTML、CSS 和 JavaScript 代码，确保 UI 在各种屏幕尺寸上都能流畅运行。

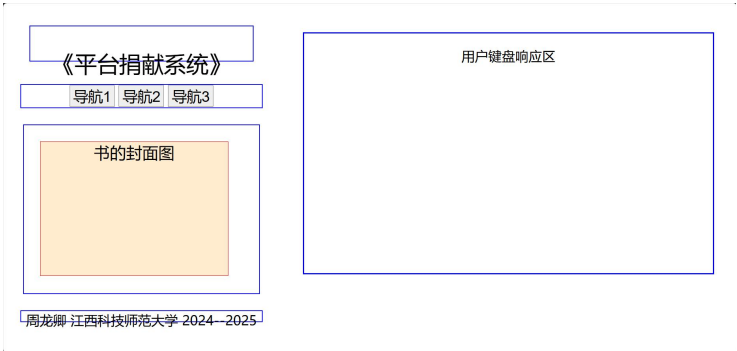


图 5.1 应用响应式设计技术开发

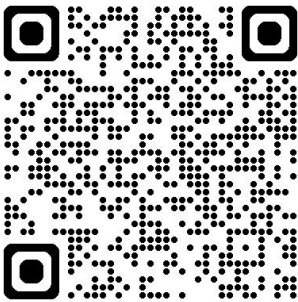


图 5.2 移动端

5.4. 项目的代码提交和版本管理

编写好的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add 1.3.html
```

成功提交代码后，gitbash 的反馈如下图 5.3 所示：

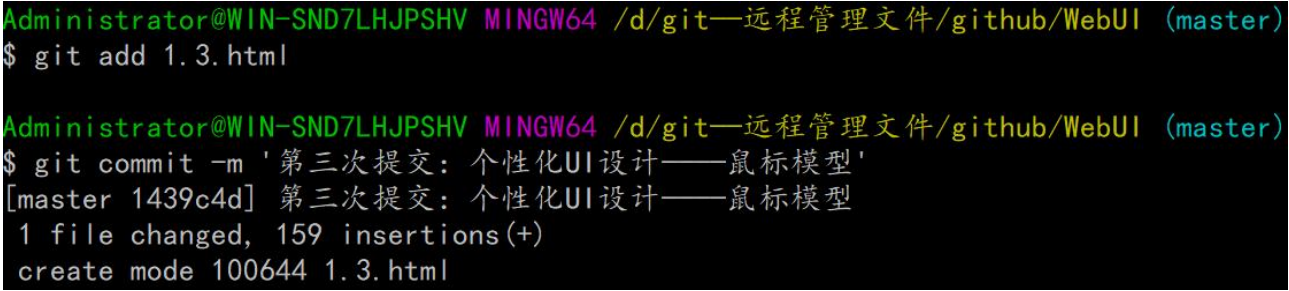


图 5.3 gitbash 提交反馈

gitbash 反馈代码的仓库日志如下所示：

```
Administrator@WIN-SND7LHJPSHV MINGW64 /d/git—远程管理文件/github/WebUI (master)
$ git log
commit 1439c4d8c9d697968fcf2186cabd849991f50c93 (HEAD -> master)
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 10:13:05 2024 +0800

    第三次提交：个性化UI设计——鼠标模型

commit e042f534996fdf254d69dbf679139b5e018fd9f9
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 09:53:59 2024 +0800

    第二次提交：移动互联时代的UI开发初步——窄屏终端的响应式设计

commit e9444c697cd8fd404c691f78abf5ce57608a4931
Author: youzelian <1431850082@qq.com>
Date:   Thu Jun 6 17:26:11 2024 +0800

    项目的第一版：
```

图 5.4 gitbash 的仓库日志反馈

6. 个性化 UI 设计——鼠标模型

6.1. 分析和设计

在 Web 应用中，鼠标作为主要的输入设备之一，对于 PC 端用户来说至关重要。设计和实现针对鼠标的个性化交互，能够提升用户体验，增强应用的易用性和吸引力。本节将详细介绍如何进行基于鼠标的个性化 UI 设计，包括分析和设计、实现和编程、运行和测试，以及代码提交和版本管理。

6.2. 项目的实现和编程

一、HTML 代码编写如下：

```
<header>
<p id="book">
    《我的毕设题目》
</p>
</header>

<nav>
<button>导航 1</button>
<button>导航 2</button>
<button>导航 3</button>
</nav>

<main id="main">
<div id="bookface">
    书的封面图
</div>
</main>

<footer>
```

CopyRight 周龙卿 江西科技师范大学 2024--2025

```
</footer>

<div id="aid">

<p>用户键盘响应区</p>

<p id="keyboard"></p>

</div>
```

二、CSS 代码编写如下:

```
*{
text-align: center;
box-sizing: border-box ;
}

header,main,div#bookface,nav,footer{
margin:1em;
}

header{
border: 2px solid blue;
height: 15%;
font-size: 1.66em;

}

main{
border: 2px solid blue;
height: 70%;
font-size: 1.2em;

}

nav{
border: 2px solid blue;
```



```
height: 10%;  
}  
nav button{  
font-size: 1.1em;  
}  
footer{  
border: 2px solid blue;  
height: 5%;  
}  
body{  
position:relative ;  
}  
#aid{  
position: absolute;  
border: 3px solid blue;  
top: 0.5em;  
left: 600px;  
}  
#bookface{  
width: 80%;  
height: 80%;  
border: 1px solid red;  
background-color: blanchedalmond;  
margin:auto;  
}
```

6.3. 项目的运行和测试

为了确保基于鼠标的个性化 UI 功能正常运行并提供优质的用户体验，编写单元测试，确保每个鼠标事件处理函数和交互逻辑都能正常工作。进行集成测试，验证各个模块之间的交互和整体功能的正确性。在不同的浏览器（如 Chrome、Firefox、Safari、Edge）上测试鼠标交互，确保一致的行为。在各种设备上测试，确保在不同屏幕尺寸和分辨率下，鼠标操作的响应和显示效果正常。鼠标模型 PC 端显示如图 5.1 所示，移动端如图 5.2 所示。

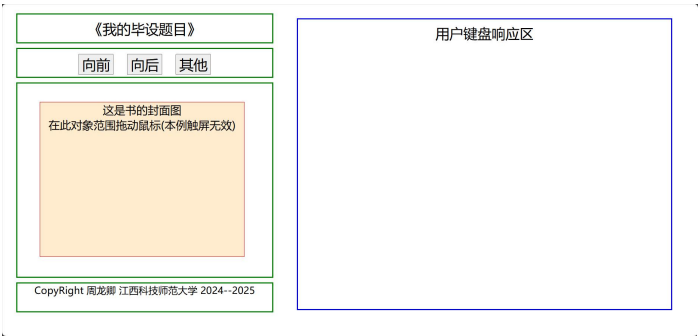


图 5.1 鼠标模型 PC 端



图 5.2 鼠标模型移动端

6.4. 项目的代码提交和版本管理

编写好的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add 1.3.html
```

成功提交代码后，gitbash 的反馈如下图 5.3 所示：

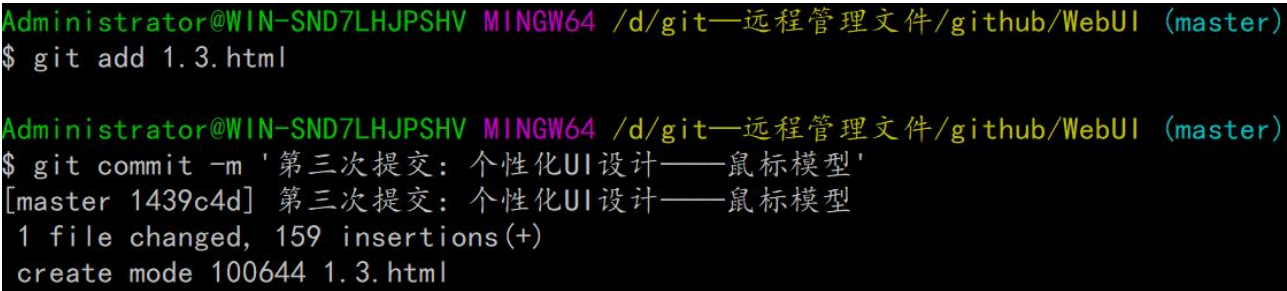


图 5.3 gitbash 提交反馈

gitbash 反馈代码的仓库日志如下所示：

```
Administrator@WIN-SND7LHJPSHV MINGW64 /d/git—远程管理文件/github/WebUI (master)
$ git log
commit 1439c4d8c9d697968fcf2186cabd849991f50c93 (HEAD -> master)
Author: youzelian <1431850082@qq.com>
Date: Wed Jun 12 10:13:05 2024 +0800

    第三次提交：个性化UI设计——鼠标模型

commit e042f534996fdf254d69dbf679139b5e018fd9f9
Author: youzelian <1431850082@qq.com>
Date: Wed Jun 12 09:53:59 2024 +0800

    第二次提交：移动互联时代的UI开发初步——窄屏终端的响应式设计

commit e9444c697cd8fd404c691f78abf5ce57608a4931
Author: youzelian <1431850082@qq.com>
Date: Thu Jun 6 17:26:11 2024 +0800

    项目的第一版：
```

图 5.4 gitbash 的仓库日志反馈

7. 对触屏和鼠标的通用交互操作的设计开发

7.1. 分析和设计

在现代 Web 应用中，用户可能通过各种设备访问，包括带有触摸屏的移动设备和使用鼠标的桌面电脑。为了提供一致的用户体验，设计和开发能够同时适应触屏和鼠标操作的通用交互界面是至关重要的。以下内容将详细探讨如何分析、设计和实现这一类交互操作。

7.2. 项目的实现和编程

一、HTML 代码编写如下：

```
<header>
<p id="book">
  《我的毕设题目》
</p>
</header>

<nav>
<button>向前</button>
<button>向后</button>
<button>其他</button>
</nav>

<main id="main">
<div id="bookface">
  这是书的封面图<br>
  在此对象范围拖动鼠标(本例触屏无效)
</div>
</main>
```

```
<footer>
```

```
CopyRight 周龙卿 江西科技师范大学 2024--2025
```

```
</footer>
```

```
<div id="aid">
```

```
<p>用户键盘响应区</p>
```

```
<p id="keyboard"></p>
```

```
</div>
```

二、CSS 代码编写如下：

```
*{  
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;
}

button{
font-size:1em;
}

#aid{
position: absolute;
border: 3px solid blue;
top:0px;
left:600px;
}

#bookface{
position: absolute;
width: 80%;
height: 80%;
border:1px solid red;
background-color: blanchedalmond;
left:7% ;
```

```
top: 7% ;  
}
```

7.3. 项目的运行和测试

在确保触屏和鼠标的通用交互操作正常运行时，应该进行全面的测试和优化：编写测试用例，确保各个交互操作在不同的输入设备上都能正常工作。进行集成测试，验证多个交互操作和模块之间的协调工作。在主流的桌面浏览器和移动浏览器中进行测试，确保一致的行为和体验。在各种设备上测试，包括不同品牌和型号的智能手机、平板电脑和 PC。邀请用户在实际设备上使用应用，观察他们的操作行为和反馈。通过问卷调查或用户访谈，收集关于交互操作的使用体验和改进建议。分析和优化事件处理的响应速度，确保在触屏和鼠标操作下都能提供流畅的体验。触屏和鼠标的通用操作 PC 端如图 6.1 所示，移动端如图 6.2 所示。

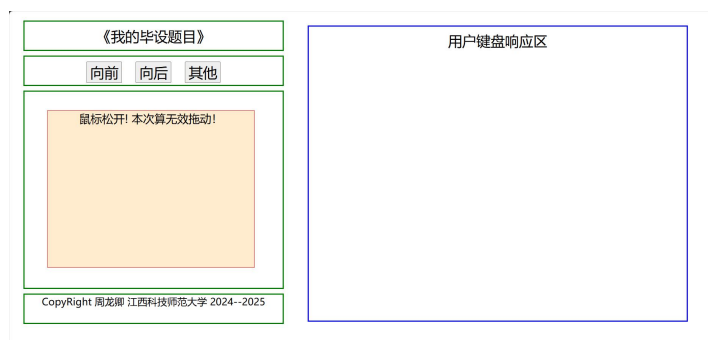


图 6.1 触屏和鼠标的通用操作 PC 端

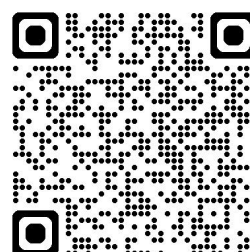


图 6.2 移动端显示

7.4. 项目的代码提交和版本管理

编写好的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add 1.4.html
```

成功提交代码后，gitbash 的反馈如下图 6.3 所示：

```
Administrator@WIN-SND7LHJPSHV MINGW64 /d/git—远程管理文件/github/WebUI (master)
$ git add 1.4.html

Administrator@WIN-SND7LHJPSHV MINGW64 /d/git—远程管理文件/github/WebUI (master)
$ git commit -m '第四次提交：对触屏和鼠标的通用交互操作的设计开发'
[master 722cd5e] 第四次提交：对触屏和鼠标的通用交互操作的设计开发
1 file changed, 190 insertions(+)
create mode 100644 1.4.html
```

图 6.3 gitbash 提交反馈

gitbash 反馈代码的仓库日志如下图 6.4 所示：

```
commit 722cd5e5c009cc0612f71daf2438e2005e16dbf2 (HEAD -> master)
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 10:20:57 2024 +0800

    第四次提交：对触屏和鼠标的通用交互操作的设计开发

commit 1439c4d8c9d697968fcf2186cabd849991f50c93
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 10:13:05 2024 +0800

    第三次提交：个性化UI设计——鼠标模型

commit e042f534996fdf254d69dbf679139b5e018fd9f9
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 09:53:59 2024 +0800

    第二次提交：移动互联网时代的UI开发初步——窄屏终端的响应式设计

commit e9444c697cd8fd404c691f78abf5ce57608a4931
Author: youzelian <1431850082@qq.com>
Date:   Thu Jun 6 17:26:11 2024 +0800

    项目的第一版：
.
```

图 6.4 gitbash 的仓库日志反馈

8. UI 的个性化键盘交互控制的设计与开发

8.1. 分析和设计

键盘交互在 Web 应用的用户体验中扮演着重要的角色，尤其是在桌面环境中。个性化的键盘交互设计不仅能提升用户操作的效率，还能为特定的应用场景提供便利^[8]。本节将详细介绍如何设计、实现和管理针对键盘的个性化 UI 交互控制。

8.2. 项目的实现和编程

一、HTML 代码编写如下：

```
<header>
<p id="book">
    《我的毕设题目》
</p>
</header>

<nav>
<button>向前</button>
<button>向后</button>
<button>其他</button>
</nav>

<main id="main">
<div id="bookface">
    这是书的封面图<br>
    在此对象范围拖动鼠标/滑动触屏<br>
    拖动/滑动超过 100 像素，视为有效 UI 互动！
</div>
</main>
```

```
<footer>
```

```
CopyRight 周龙卿江西科技师范大学 2024--2025
```

```
</footer>
```

```
<div id="aid">
```

```
<p>用户键盘响应区</p>
```

```
</div>
```

二、CSS 代码编写如下：

```
* {  
  
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;
}

button {
font-size: 1em;
}

#aid {
position: absolute;
border: 3px solid blue;
top: 0px;
left: 600px;
}
```

```
#bookface {
position: absolute;
width: 80%;
height: 80%;
border: 1px solid red;
background-color: blanchedalmond;
left: 7%;
top: 7%;
}
```

8.3. 项目的运行和测试

UI 个性化键盘交互控制的 PC 端如图 7.1 所示，移动端如图 7.2 所示。



图 8.1 UI 个性化键盘交互控制 PC 端

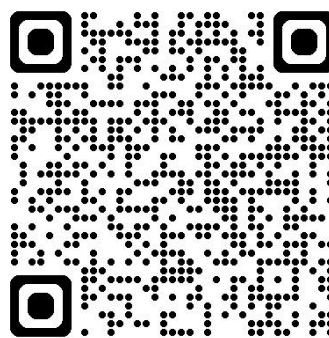


图 8.2 移动端访问

8.4. 项目的代码提交和版本管理

编写好的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add 1.5.html
```

成功提交代码后，gitbash 的反馈如下图 7.3 所示：

```
Administrator@WIN-SND7LHJPSHV MINGW64 /d/git—远程管理文件/github/WebUI (master)
$ git add 1.5.html

Administrator@WIN-SND7LHJPSHV MINGW64 /d/git—远程管理文件/github/WebUI (master)
$ git commit -m '第五次提交: UI的个性化键盘交互控制的设计与开发'
[master 3d56481] 第五次提交: UI的个性化键盘交互控制的设计与开发
1 file changed, 225 insertions(+)
create mode 100644 1.5.html
```

图 8.3 gitbash 提交反馈

gitbash 反馈代码的仓库日志如下图 7.4 所示:

```
Administrator@WIN-SND7LHJPSHV MINGW64 /d/git—远程管理文件/github/WebUI (master)
$ git log
commit 3d564816fe618b837025fb0cd65474427e15c62b (HEAD -> master)
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 10:26:29 2024 +0800

    第五次提交: UI的个性化键盘交互控制的设计与开发

commit 722cd5e5c009cc0612f71daf2438e2005e16dbf2
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 10:20:57 2024 +0800

    第四次提交: 对触屏和鼠标的通用交互操作的设计开发

commit 1439c4d8c9d697968fcf2186cabd849991f50c93
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 10:13:05 2024 +0800

    第三次提交: 个性化UI设计——鼠标模型

commit e042f534996fdf254d69dbf679139b5e018fd9f9
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 09:53:59 2024 +0800
```

图 8.4 gitbash 日志反馈

9. 谈谈本项目的高质量代码

在 Web 流程中，质量代码的编写至关重要，不仅要考虑项目的性能和可维护性，还要考虑团队协作和未来扩展的可能性。以下是本项目在实现质量代码方面的主要策略和实践。

9.1. 清晰的错误结构和注释

清晰的代码结构是高质量代码的基础。在本系统中，我们严格遵循模块化设计原则，将功能相似的代码分组到嵌套模块中。这种做法不仅提高了代码的可读性，还便于维护和调试。每个模块都有均匀的职责分工，遵循“单一职责原则”，确保每个模块只负责特定的功能，避免代码的耦合和复杂度。

此外，我们在代码中广泛使用注释，关注逻辑复杂或关键部分。注释帮助任务负责人理解代码的功能，理解和修改代码的难度。在 JavaScript 代码中，使用 JSDoc 风格的注释，为函数和对象提供详细描述和使用示例，并且会自动生成 API 文档，并提高代码的自我解释性。

9.2. 代码规范和标准

代码规范是保证代码符合规范的关键。本项目需遵循行业标准代码风格指南，例如 Airbnb 的 JavaScript 编码规范，以及 ESLint 和 Prettier 自动检查和格式化代码。这些工具可以帮助保持代码的规范性，避免因个人编码习惯差异而导致代码风格混乱。

我们在 CSS 和 HTML 的编写中也遵循了类似的规范。使用 BEM（Block, Element, Modifier）命名约定，使得 CSS 类名具有良好的可读性和可维护性。同时，保持 HTML 结构的简洁和语义化，提高了代码的可理解和 SEO（搜索引擎优化）效果。

9.3. 代码的可测试性

该项目旨在为我们提供编写脚本和集成平台，从而保证每个功能模块的完整性和可执行性。该项目采用了 Jest 作为我们的代码执行引擎，它提供了强大的代码执行能力和覆盖率。每次执行完代码后，我们就可以生成完整的代码，确保所有功能模块的完整性和可执行性。

对于 UI 组件，我们使用了 React Testing Library 和 Cypress 进行 UI 的单元测试和端到端测试。这些工具可以帮助我们模拟用户交互，验证 UI 的各种程度行为，并保证用户界面的可用性。

9.4. 代码性能优化

在本项目中，我们通过了几种主要方法和流程改进了 Web 应用性能：

- 懒加载 (Lazy Loading)：对于大型资源并且不立即需要的组件，使用懒加载技术，延迟它们的加载时间，从而加快初始页面的加载速度。
- 代码拆分 (Code Splitting)：利用 Webpack 等工具，我们将代码拆分成多个小块，使得浏览器可以包含加载这些块，构建加载时间和内存占用。
- 缓存：使用浏览器缓存服务和缓存技术，避免重复加载静态资源，并提高页面的响应速度。
- 压缩和压缩：通过对 JavaScript、CSS 和图像文件进行压缩和压缩，解析文件的大小，提高了传输效率。

9.5. 代码版本控制

为了有效管理代码变更，使用了 Git 进行版本控制。每个功能和修复都通过编译分支进行开发，并在合并到主分支之前经过代码和测试。这种练习不仅提高了代码的质量，还确保了团队能够协同工作，以及合并冲突和代码回归研究。

通过这些措施，我们在本地区实现了全部工作量，保证了 Web 应用的可靠性、可重复性和可用性，为用户提供了优质的用户体验。这些措施不仅适用于当前项目，也为未来的 Web 开发项目提供了价值参考和指导。

10.用 gitBash 工具管理项目的代码仓库和 http 服务器

10.1. 跨世纪的经典 Bash 工具

Bash (Bourne Again SHell)是由 GNU 项目开发的一个广泛应用于 Unix 和类 Unix 操作系统的命令行解释器和脚本语言。自 1989 年发布以来, Bash 已成为系统管理员、开发者和高级用户进行系统管理和脚本编程的首选工具。它融合了早期 Unix shell 的功能, 并进行了大量增强, 使得 Bash 成为一个跨世纪的经典工具^[9]。

Bash 是一款既可以作为命令行解释器, 也可以作为脚本编程语言的工具。在 Unix 系统中, shell 是用户与操作系统之间的接口, 它允许用户输入命令并得到系统的响应。Bash 作为一个 shell, 它不仅支持基本的命令执行, 还提供了强大的脚本编写功能。这使得用户可以编写脚本来自动化日常任务, 如文件管理、系统维护和软件安装等。

Bash 的名字来源于 Bourne shell 的增强版本 ("Bourne Again SHell")。Bourne shell (sh)是由斯蒂芬·布尔恩在 1970 年代开发的, 它是 Unix 操作系统的标准 shell。Bash 继承了 Bourne shell 的许多特性, 并在此基础上进行了扩展。它的发布不仅标志着 Unix shell 的一个重大进步, 也成为了 GNU 操作系统的重要组成部分。在过去的几十年里, Bash 不断发展和演变。它的兼容性和功能性使其成为 Linux 和 macOS 等系统的默认 shell。随着版本的升级, Bash 不断引入新的功能和改进, 如数组支持、改进的错误处理、增强的字符串操作等。

在现代计算环境中, Bash 仍然发挥着重要作用。虽然图形用户界面 (GUI) 变得越来越流行, 但命令行界面 (CLI) 在许多任务中仍然具有无可替代的效率和灵活性。Bash 提供的强大命令行功能和脚本能力, 使其成为自动化、系统管理和开发工作的重要工具。特别是在 DevOps 和云计算领域, Bash 脚本被广泛用于配置管理、部署自动化和系统监控等任务。无论是编写启动脚本, 还是配置服务器环境, Bash 都是开发者和系统管理员的得力助手。对于 Windows 用户来说, GitBash 是一个将 Bash 的强大功能带到 Windows 平台的重要工具。它

集成了 Git 和 Bash，提供了一个仿 Linux 的环境，使得 Windows 用户能够享受到 Bash 的强大功能。GitBash 使得开发者可以在 Windows 系统上方便地使用 Bash 命令和脚本，同时也可以管理 Git 仓库，进行版本控制^[10]。

10.2. 利用 gitBash 工具对本项目进行管理

10.2.1. 项目文件的基本搭建

第一步，建立项目的基本文件结构，并设置好代码仓库的基本信息，包括开发者的名字和电子邮箱等。具体操作步骤如下，通过 gitBash 命令行执行以下命令（如所示）。

```
$ cd /d/git——远程管理文件/github
$ mkdir WebUI
$ cd WebUI
$ git init
$ git config user.name 江科大余泽亮
$ git config user.email 1431850082@qq.com
$ touch index.html myCss.css myJs.js
```

10.2.2. 项目代码的提交与上传

在编写并成功运行 index.html、myCss.css 和 myJs.js 的代码后，执行以下操作（）。其中，成功提交文件后，gitBash 的反馈如图所示。

```
$ git add index.html myCss.css myJs.js
$ git commit -m '项目第一版：”三段论”式的内容设计概要'
```

```
Administrator@WIN-SND7LHJPSHV MINGW64 /d/git——远程管理文件/github/WebUI (master)
$ git add index.html myCSS.css myJs.js

Administrator@WIN-SND7LHJPSHV MINGW64 /d/git——远程管理文件/github/WebUI (master)
$ git commit -m '项目第一版：三段论式的内容设计概要'
[master 5da63d8] 项目第一版：三段论式的内容设计概要
1 file changed, 351 insertions(+)
create mode 100644 myJs.js
```

图 8.1 gitbash 代码提交反馈

10.2.3. 项目日志的反馈查询

在项目开发过程中，使用 `git log` 命令查看和管理 Git 的提交日志是很重要的。仓库日志如图 8.2 所示。

```
Administrator@WIN-SND7LHJPSHV MINGW64 /d/git—远程管理文件/github/WebUI (master)
$ git log
commit 3d564816fe618b837025fb0cd65474427e15c62b (HEAD -> master)
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 10:26:29 2024 +0800

    第五次提交：UI的个性化键盘交互控制的设计与开发

commit 722cd5e5c009cc0612f71daf2438e2005e16dbf2
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 10:20:57 2024 +0800

    第四次提交：对触屏和鼠标的通用交互操作的设计开发

commit 1439c4d8c9d697968fcf2186cabd849991f50c93
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 10:13:05 2024 +0800

    第三次提交：个性化UI设计——鼠标模型

commit e042f534996fdf254d69dbf679139b5e018fd9f9
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 09:53:59 2024 +0800

    第二次提交：移动互联时代的UI开发初步——窄屏终端的响应式设计
... skipping ...
commit 3d564816fe618b837025fb0cd65474427e15c62b (HEAD -> master)
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 10:26:29 2024 +0800

    第五次提交：UI的个性化键盘交互控制的设计与开发
```

图 8.2 gitbash 仓库日志反馈

10.3. 通过 GitHub 平台实现本项目的全球域名

通过 GitHub 平台实现本项目的全球域名 GitHub 是全球最受欢迎的代码托管平台之一，不仅提供强大的版本控制和协作工具，还为开发者提供便捷的方式来托管静态网站——这就是 GitHub Pages。通过 GitHub Pages，您可以将项目托管在 GitHub 上，并且获得一个免费且全球可访问的域名，使项目能够被他人访问和分享。

设置本地仓库与远程代码仓库的连接并推送

创建好仓库后，我们需要将本地与 GitHub 创建的仓库进行连接。具体命令如下所示。如果出现如图 8.4 所示的情况，则表明成功了。

```
$ git init
```

```
$ git remote add origin https://github.com/youzelian/WebUI.git
```

```
Administrator@WIN-SND7LHJPSHV MINGW64 /d/git—远程管理文件/github/WebUI
$ git init
Initialized empty Git repository in D:/git—远程管理文件/github/WebUI/.git/

Administrator@WIN-SND7LHJPSHV MINGW64 /d/git—远程管理文件/github/WebUI (master)
$ git remote add origin https://github.com/youzelian/WebUI.git
```

图8.4 本地仓库与远程仓库连接

参考文献

- [1] 刘华星, 杨庚. HTML5——下一代 Web 开发标准研究[J]. 计算机技术与发展, 2011,21(08):54-58.
- [2]HTML5在移动应用开发上的应用前景 {Author}:黄永慧;陈程凯[J]. 计算机技术与发展, 2013,23 {Issue}: 07:207-210.
- [3]基于响应式Web设计的网页模板的设计与实现 {Author}: 张树明[J]. 计算机与现代化, 2013,{Issue}: 06:125-127.
- [4]UML建模技术及其应用 {Author}: 陆晓燕,秦朝辉,尹治本[J]. 成都信息工程学院学报, 2004:414-417.
- [5]软件开发模型研究综述 {Author}:张友生;李雄[J]. 计算机工程与应用, 2006:109-115.
- [6]瀑布模型、快速原型模型和增量模型的对比 {Author}: 张越[J]. 电子技术与软件工程, 2019,{Issue}: 03:32.
- [7]Web服务核心支撑技术:研究综述 {Author}: 岳昆,王晓玲,周傲英[J]. 软件学报, 2004:428-442.
- [8]罗超. 鼠标、键盘退场智能硬件掀起交互革命[J]. IT时代周刊, 2014(10):36-37.
- [9]郑桦. 嵌入式Linux文件系统的设计与实现[D]. 武汉理工大学, 2004.
- [10]Git——版本管理之利器 {Author}: 宋冬生[J]. 程序员, 2007,{Issue}: 11:118-119.