

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：张子然

学 号：20213614

指导教师：李建宏

2024 年 6 月 12 日

Web 技术的个性化 UI 设计	1
1. 引言	2
1.1 背景与意义	2
1.2 国内外课题研究现状	2
1.3 项目概述	3
2. 技术总结和文献综述	4
2.1 技术总结	4
2.2 文献综述	5
3. 内容设计概要	6
3.1 项目分析和设计	6
3.2 项目的实现和编程	6
3.3 项目的运行和测试	8
3.4 项目的代码提交和版本管理	8
4. 移动互联时代的 UI 开发初步——窄屏终端的响应式设计	10
4.1 项目分析和设计	10
4.2. 项目的实现和编程	10
4.3 项目的运行和测试	12
1.1. 项目的代码提交和版本管理	13
5. 应用响应式设计技术开发可适配窄屏和宽屏的 UI	14
5.1 分析和设计	14
5.2 项目的实现和编程	14
5.3 项目的运行和测试	16

5.2 项目的代码提交和版本管理	17
6. 个性化 UI 设计——鼠标模型	18
6.1 分析和设计	18
6.2 项目的实现和编程	18
6.3 项目的运行和测试	21
6.4 项目的代码提交和版本管理	21
7. 对触屏和鼠标的通用交互操作的设计开发	23
7.1 分析和设计	23
7.2 项目的实现和编程	23
7.3 项目的运行和测试	26
7.4 项目的代码提交和版本管理	27
8. UI 的个性化键盘交互控制的设计与开发	29
8.1 分析和设计	29
8.2 项目的实现和编程	29
8.3 项目的运行和测试	32
8.4 项目的代码提交和版本管理	33
9. 谈谈本项目中的高质量代码	35
9.1 响应式设计	35
9.2 键盘导航优化	35
9.3 键盘输入实时反馈	35
9.4 自定义快捷键	35
9.5 视觉与交互的完美结合	35

10. 用 gitBash 工具管理项目的代码仓库和 http 服务器	37
10.1 跨世纪的经典 Bash 工具	37
10.2 利用 gitBash 工具对本项目进行管理	37
10.3 通过 GitHub 平台实现本项目的全球域名	38

Web 技术的个性化 UI 设计

摘要: 本文在现代数字化和移动互联的背景下, 重点探讨了基于 Web 客户端技术的个性化与响应式用户界面 (UI) 设计与开发的重要性。研究围绕 Web 平台的核心技术, 如 HTML5、CSS3 和 JavaScript, 详细探讨了这些技术在 UI 设计中的实际应用。通过 DOM 技术和事件驱动模式, 项目成功实现了对鼠标、触屏和键盘等输入设备的底层事件响应和流畅支持, 并创新性地设计了一个对象模型来模拟这些指向性设备, 提升了交互的灵活性和准确性。此外, 项目还引入响应式设计概念, 结合媒体查询和弹性布局技术, 确保 Web 应用能在不同设备 (从智能手机到桌面显示器) 上实现自动适应, 为用户提供一致且优化的体验。同时, 文章深入分析了如何利用用户数据和行为模式进行个性化 UI 设计, 以满足不同用户的操作习惯和视觉偏好, 进一步增强了用户体验的个性化与满意度。在项目管理方面, 项目采用增量式迭代开发模型, 结合 GitBash 和 GitHub 工具进行版本控制和全球协作开发, 有效提升了开发效率和代码质量。通过实际案例的实施、测试和验证, 本研究充分证明了基于 Web 技术的个性化 UI 设计在提升用户体验和适应多样化设备需求方面的显著效果, 为 Web 应用的未来发展提供了宝贵的参考和启示。

关键词: Web 客户端技术、Git 工具与版本控制、HTML5/CSS3/JavaScript、DOM 技术与事件驱动。

Personalized UI design for Web technology

Abstract: In the context of modern digitalization and mobile Internet, this paper focuses on the importance of personalized and responsive user interface (UI) design and development based on web client technology. The study revolves around the core technologies of the web platform, such as HTML5, CSS3, and JavaScript, and explores in detail the practical applications of these technologies in UI design. Through DOM technology and event-driven mode, the project successfully implemented low-level event response and fluent support for input devices such as mouse, touch screen, and keyboard, and innovatively designed an object model to simulate these directional devices, improving the flexibility and accuracy of interactions. In addition, the project introduced a responsive design concept, combined with media queries and flexible layout techniques, to ensure that web applications automatically adapt across different devices, from smartphones to desktop displays, to provide users with a consistent and optimized experience. At the same time, this paper deeply analyzes how to use user data and behavior patterns to personalize UI design to meet the operating habits and visual preferences of different users, and further enhances the personalization and satisfaction of user experience. In terms of project management, the project adopts an incremental iterative development model, combined with GitBash and GitHub tools for version control and global collaborative development, which effectively improves development efficiency and code quality. Through the implementation, testing and verification of practical cases, this study fully proves the remarkable effect of personalized UI design based on Web technology in improving user experience and adapting to the needs of diverse devices, and provides valuable reference and inspiration for the future development of Web applications.

Key words: Web client technologies, Git tools and version control, HTML5/CSS3/JavaScript, DOM technologies and event-driven

1. 引言

1.1 背景与意义

在数字化和移动互联的时代，用户界面（UI）设计已成为 Web 开发中不可或缺的一环。随着技术的飞速发展和用户需求的多样化，个性化 UI 设计逐渐成为提升用户体验、增强产品竞争力的关键。本项目致力于基于 Web 客户端技术开发一个个性化的用户界面，以满足不同用户在不同设备上的需求，提供流畅、一致的交互体验。

1.2 国内外课题研究现状

1.2.1 国内研究现状

在国内，HTML5、CSS3、JavaScript 等现代 Web 技术得到了广泛应用，特别是在响应式设计和个性化 UI 设计方面。这些技术的应用使得 UI 设计更加灵活、多样化，并能够更好地满足用户需求。同时，国内设计师和团队也在不断探索新的设计理念和交互方式，如利用 AI 技术进行智能推荐、数据分析等，以提升用户体验和产品竞争力。尽管国内 UI 设计行业取得了显著进展，但仍面临一些挑战和问题。首先，设计师技能水平参差不齐，缺乏统一的设计规范和标准。这导致了一些 UI 设计作品质量不高、用户体验不佳。其次，随着移动互联网的普及和多样化设备的出现，如何确保 UI 设计在不同设备和屏幕尺寸上提供一致、流畅的交互体验成为了一个重要问题。

1.2.2 国外相关研究

在国外，特别是在发达国家，Web 开发技术已经相当成熟，HTML5、CSS3、JavaScript 等技术的应用已经普及到各个领域。同时，国外 UI 设计行业也更加注重技术应用的标准化和规范化，以确保设计作品的质量和一致性。

国外 UI 设计行业更加注重用户体验研究，通过深入分析用户行为和需求，提供更加符合用户习惯和偏好的设计方案。同时，国外设计师和团队也在不断探索新的交互方式和设计理念，以提供更加创新、便捷的用户体验。在跨平台和多设备兼容方面，国外 UI 设计行业也取得了显著进展。通过采用响应式设计和动态布局等技术手段，设计师能够确保 UI 设计在不同设备和屏幕尺寸上提供一致、流畅的交互体验。这有助于提升产品的市场竞争力并吸引更多用户。

1.3 项目概述

在现代 Web 开发中，用户界面（UI）设计是至关重要的一部分。随着用户需求的多样化和技术的迅猛发展，个性化 UI 设计已经成为提升用户体验的关键因素。个性化 UI 不仅能够根据用户的偏好和行为进行调整，还能够不同设备和屏幕尺寸上提供一致且流畅的交互体验。

本项目的主要目标是通过利用现代 Web 技术，如 HTML5、CSS3、JavaScript 等，以及响应式设计原则，实现一个具备以下特性的个性化用户界面：

1. 响应式设计：设计并实现能够自适应不同屏幕尺寸（如手机、平板和桌面设备）的用户界面，确保在各种设备上提供一致、可用的用户体验。

2. 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，为用户提供更加个性化的服务。

3. 多种输入支持：开发支持鼠标、键盘和触屏的通用交互模式，确保用户能够在不同设备上流畅、便捷的操作。

4. 高质量代码与版本管理：采用良好的编码规范和版本控制工具（如 Git），确保代码的可维护性、可扩展性和团队协作效率。

通过这个项目，我们不仅希望展示现代 Web 技术在个性化 UI 设计中的应用，还希望探索和解决在多设备、多交互模式下的设计和开发挑战。最终，项目将为开发人员提供一个参考案例，展示如何利用先进的 Web 客户端技术实现高质量、个性化的用户界面。

2. 技术总结和文献综述

2.1 技术总结

2.1.1 HTML5

HTML5 (HyperText Markup Language 5) 即超文本标记语言技术, 是万维网上主要的标记语言标准。在 HTML5 技术中, 它不仅包括所有 HTML 技术, 还增加了 JavaScript 脚本语言和 CSS 层叠样式。相对而言, 在网页开发语言中使用 HTML5 技术可以简化网页的结构, 增加浏览器用户的交互性, 减少浏览器对相关插件的依赖。HTML5 还具有低成本开发、支持多设备跨平台应用、支持实时更新功能等应用优势。HTML5 技术对于 Web 前段开发设计的可行性较高, HTML5 技术的设计旨在在移动设备上支持多媒体应用。就 HTML5 技术的应用优势来看, 其具备很多新功能, 还能满足实时更新需求, 能够有效改变用户和文档交互方式, 尤其是在移动设备终端上的 Web 应用程序开发中, 其应用优势比较突出。在具体的 Web 前端开发中, 使用 HTML5 技术, 可以让开发者的前端开发工作量减少, 开发的工作效率显著提升, 这对于互联网领域的发展而言也有里程碑式的意义^[1]。

2.1.2 JavaScript

行为即是交互逻辑, Web 开发中大部分交互逻辑都是由 JavaScript 实现, 它可以直接内嵌于 HTML 网页, 也可以写成有利于结构和行为的分离单独的 JavaScript 文件, 通过浏览器内置的 JavaScript 引擎直接编译, 把一个原来只用来显示的页面, 转变成支持用户交互的页面程序。JavaScript 主要运作于浏览器当中, 浏览器主要由渲染引擎和 JavaScript 引擎组成, 渲染引擎俗称内核, 以谷歌浏览器为例, blink 用来解析 HTML 与 CSS 代码。JavaScript 引擎也称 JavaScript 解释器, 比如谷歌浏览器的 V8 用来读取网页中的 JavaScript 代码, 对解析处理后运行。JavaScript 诞生于 1995 年, 由网景公司的 Brendan Eich 设计, 是一种运行在客户端的脚本语言, 在设计之初, JavaScript 一般用于浏览器表单数据校验, 在 Web 开发实际过程中, JavaScript 也可基于语言编程规范, 强化网页数据安全, 确保信息有效性。JavaScript 主要由 ECMAScript、DOM、BOM 三个部分组成, 其中 ECMAScript 指的就是 JavaScript 的语法规则。DOM 指的是文档对象模型, 描述处理网页内容的方法和接口。BOM 是浏览器对象模型, 即浏览器进行交互的方法和接口^[2]。

2.1.3 CSS3

CSS3 技术是 CSS 技术的最新一次,也是所谓的分层技术^[3],它可以将 W3C 分成若干个不同的模块。与其他技术相比,CSS3 技术能在网页的前端版式设计中,使网页呈现多种版式,例如,正面圆角效果、界面边缘图形模糊化、文字印象效果、透明效果、渐变效果等。采用 Font-Face 技术,可以自定义页面的前端和各种动作处理,包括移动、旋转、放大、缩小等,使其更多样化。CSS 是分层的风格,也是一种不需要编辑就能从浏览器中直接运行的选项,可以管理网页的界面。

2.2 文献综述

随着互联网的快速发展和普及,用户界面(UI)设计在软件开发中扮演着越来越重要的角色。现代 Web 开发技术的不断进步为 UI 设计提供了更多的可能性和创新空间。随着 HTML5 和 CSS3 的广泛应用,HTML5 和 CSS3 将给 Web 带来巨大变化。例如,邹晓丹(2023)指出了 HTML5 和 CSS3 对网页前端的设计优化,并表明未来,HTML5+CSS3 将会是一个被大量推广的技术,它将会为 Web 前端标准化和优化设计提供更为可靠的技术支持。

JavaScript 及其框架的发展同样对 UI 设计产生了深远影响。正如许至晶 2022 年《JavaScript 在 Web 开发的应用研究》所说的 JavaScript 作为嵌入式的高级程序语言,是 Web 开发中一项不可或缺的技术,大大提高 Web 开发应用功能。

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。

总体而言,现代 Web 开发技术在 UI 设计中的应用与发展为 UI 设计领域带来了更多的可能性和创新空间。通过响应式设计、个性化 UI 设计和丰富的交互效果等方法,现代 Web 开发技术使得 UI 设计更加灵活、多样化和个性化,提高了用户体验和产品竞争力。未来,随着技术的不断进步和创新,现代 Web 开发技术将继续为 UI 设计领域带来更多的机遇和挑战。我们期待更多的设计师和开发者能够积极探索和应用现代 Web 开发技术,推动 UI 设计领域的不断发展和进步。

3. 内容设计概要

3.1 项目分析和设计

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 **Error! No text of specified style in document..1** 所示：

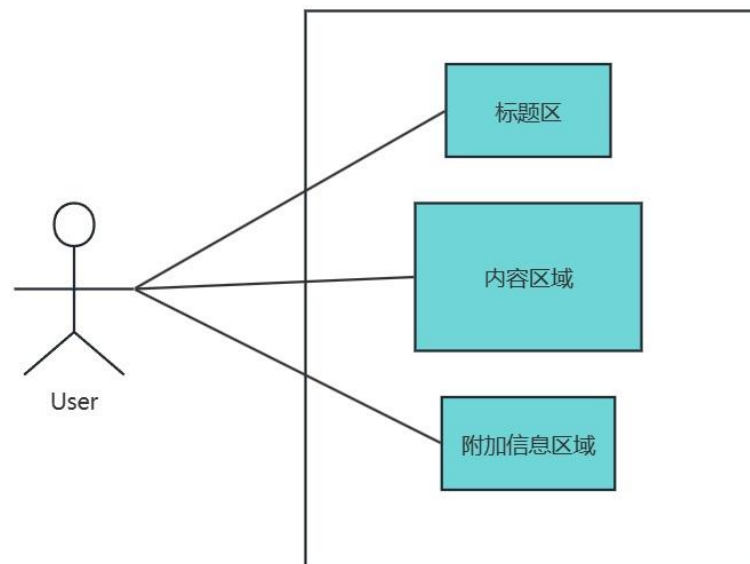


图 **Error! No text of specified style in document..1** 用例图

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

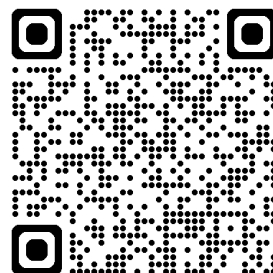


图 3.3 移动端二维码

3.4 项目的代码提交和版本管理

为了启动本项目的第一次迭代，我们计划利用 GitBash 工具来管理项目的文件。在代码提交和版本控制的关键阶段，我们的首要任务是确立项目的基础文件结构，并配置代码仓库的基本信息，包括参与开发者的姓名和电子邮件地址。这些设置将为我们后续的协作开发奠定坚实的基础，确保代码的版本追踪和变更管理能够高效进行^[4]。进入 gitBash 命令行后，按次序输入以下命令：

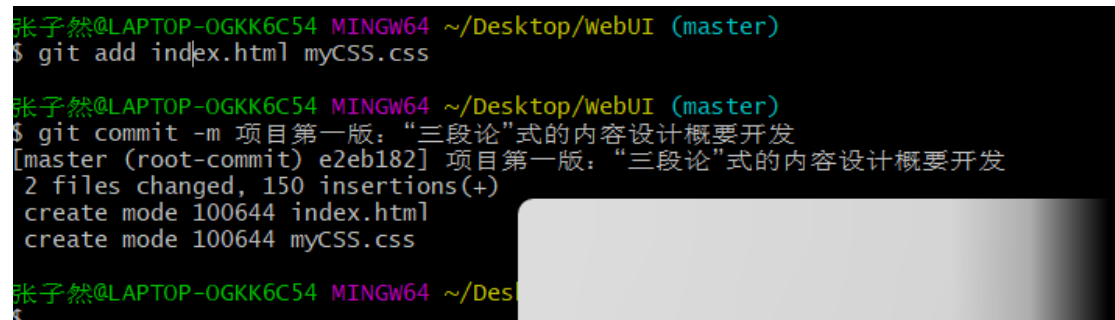
```
$ cd /  
$ mkdir webUI  
$ cd webUI  
$ git init  
$ git config user.name 江科师大张子然
```

```
$ git config user.email 2292074611@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。



```
张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)
$ git add index.html myCSS.css

张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)
$ git commit -m 项目第一版：“三段论”式的内容设计概要开发
[master (root-commit) e2eb182] 项目第一版：“三段论”式的内容设计概要开发
2 files changed, 150 insertions(+)
create mode 100644 index.html
create mode 100644 myCSS.css

张子然@LAPTOP-OGKK6C54 MINGW64 ~/Des
```

图 Error! No text of specified style in document..4 gitbash 代码提交反馈

项目代码仓库自此也开启了严肃的历史记录，我们可以输入日志命令查看，

```
$ git log
```

gitbash 反馈代码的仓库日志如下图 3.5 所示：



```
张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)
$ git log
commit e2eb1823939709aa58af8fba4e4feda6e4ae7b64 (HEAD -> master)
Author: ZZR229 <2292074611@qq.com>
Date: Tue Jun 18 20:26:00 2024 +0800

    项目第一版：“三段论”式的内容设计概要开发

张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)
$ |
```

图 Error! No text of specified style in document..5 gitbash 的仓库日

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 项目分析和设计

随着移动设备的广泛普及和网络技术的迅猛进步，如今用户越来越倾向于通过不同尺寸和类型的设备来访问 Web 应用。在这个移动优先的时代，窄屏终端，如智能手机和平板电脑，已成为用户日常网络活动的重要载体。为了应对这一趋势，响应式设计技术应运而生，它通过灵活调整用户界面（UI）的设计以适应不同屏幕尺寸，确保 Web 应用在各种设备上都能提供一致且优质的用户体验。这种设计方式不仅提升了 Web 应用的可用性，也增强了用户的满意度和忠诚度。项目的实现和编程

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">
XXX @ 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

在推进响应式设计的过程中，确保项目在不同设备和浏览器上具备卓越的兼容性和性能是至关重要的。我们需要在主流的浏览器（如 Chrome、Firefox、Safari、Edge）上进行详尽的 UI 测试，以确保无论用户选择哪种浏览器，都能获得一致的视觉呈现和交互体验。此外，我们还需要在真实的移动设备上对 UI 进行严格的测试，以验证其在不同屏幕尺寸和分辨率下的表现，确保在各种移动设备上都能提供优质的用户体验。为了快速验证设计在不同设备上的效果，我们可以利用开发者工具（如 Chrome DevTools）中的设备模拟功能进行高效的测试。同时，我们还将使用专业的性能分析工具（如 Lighthouse 或 PageSpeed Insights）来评估页面的性能指标，包括加载时间、交互延迟和渲染速度等，以便及时发现并解决潜在的性能问题。为了进一步提升应用的响应速度，我们将实施代码分割和异步加载等优化策略，确保用户能够更快地访问和使用应用。此外，我们还将通过用户测试来收集实际用户的反馈，深入了解 UI 在不同设备上的使用体验，并根据反馈进行针对性的调整和改进，以不断提升用户满意度。其中，PC 端口如图 4.1 所示，移动端访问如图 4.2 所示。

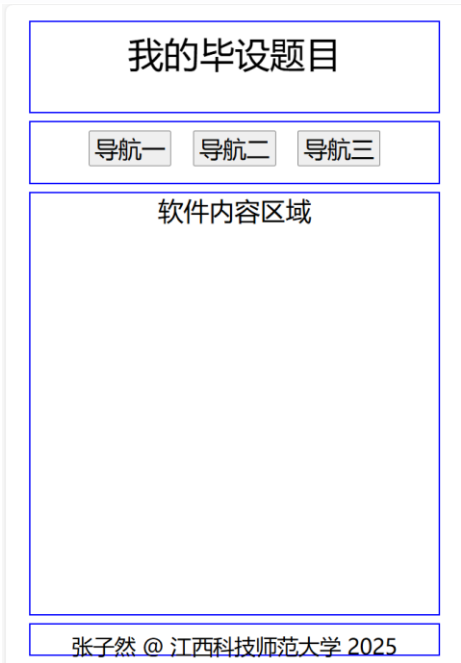


图 4.1 移动端项目的 PC 端访问

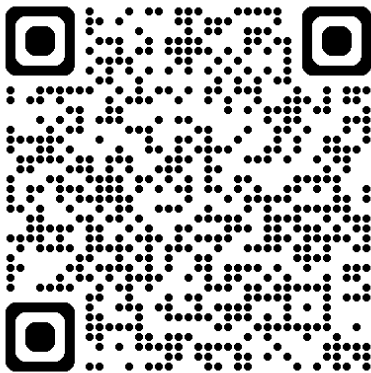


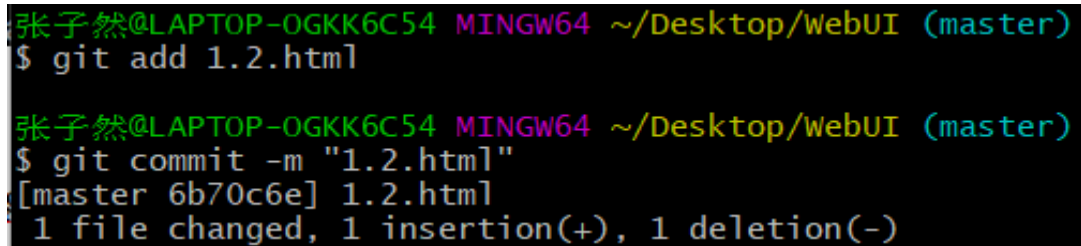
图 4.2 移动端访问

1.1. 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  
$ git commit -m "1.2.html"
```

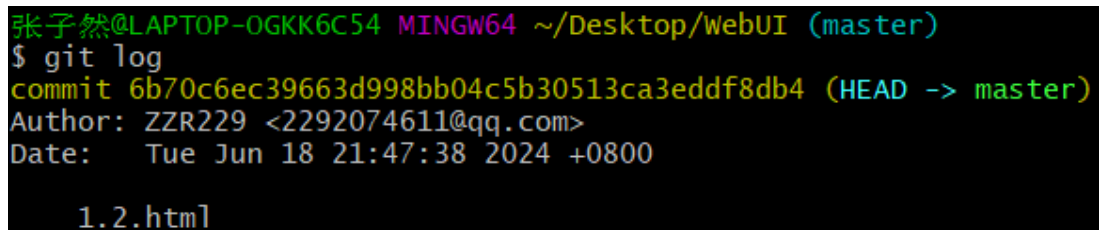
成功提交代码后，gitbash 的反馈如下图 4.3 所示：



```
张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)  
$ git add 1.2.html  
  
张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)  
$ git commit -m "1.2.html"  
[master 6b70c6e] 1.2.html  
1 file changed, 1 insertion(+), 1 deletion(-)
```

图 4.3 gitbash 提交反馈

gitbash 反馈代码的仓库日志如下所示：



```
张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)  
$ git log  
commit 6b70c6ec39663d998bb04c5b30513ca3eddf8db4 (HEAD -> master)  
Author: ZZR229 <2292074611@qq.com>  
Date: Tue Jun 18 21:47:38 2024 +0800  
  
1.2.html
```

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

应用响应式设计技术开发可适配窄屏和宽屏的 UI 是一个复杂但至关重要的过程，它确保了用户无论使用何种设备或屏幕尺寸，都能获得一致且优化的体验。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。

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>

CopyRightXXX 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;  
}
```

5.3 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 端如图 5.1 所示，移动端访问如图 5.2

所示。

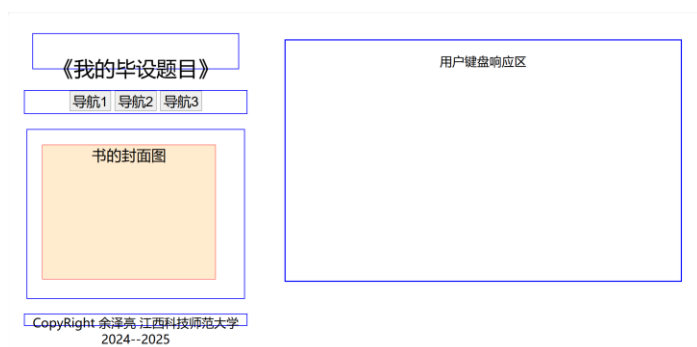


图 5.1 PC 端

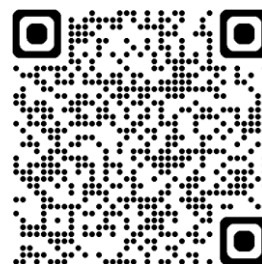


图 5.2 移动端

5.2 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  
$ git commit -m "1.3.html"
```

成功提交代码后，gitbash 的反馈如下图 5.3 所示：

```
张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)  
$ git add 1.3.html  
  
张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)  
$ git commit -m "1.3.html"  
[master 98b588c] 1.3.html  
1 file changed, 159 insertions(+)  
create mode 100644 1.3.html
```

图 5.3 gitbash 提交反馈

gitbash 反馈代码的仓库日志如下所示：

```
张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)  
$ git log  
commit 98b588c5b77f496514d8652cc452a08e339b2557 (HEAD -> master)  
Author: ZZR229 <2292074611@qq.com>  
Date: Tue Jun 18 22:01:10 2024 +0800  
  
1.3.html  
  
commit 6b70c6ec39663d998bb04c5b30513ca3eddf8db4  
Author: ZZR229 <2292074611@qq.com>  
Date: Tue Jun 18 21:47:38 2024 +0800  
  
1.2.html
```

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

在构建现代化的 Web 应用时，鼠标作为 PC 用户日常操作中的核心输入设备，其交互体验的优化对于提升整体用户体验、增强应用的易用性和吸引力至关重要。为了实现这一目标，我们需要精心策划和实施基于鼠标的个性化用户界面（UI）设计。这一流程涵盖了对用户行为的深入分析、设计理念的明确、实际开发编程、应用运行与测试，以及最终代码的提交与版本管理。通过这一系列步骤，我们能够确保 Web 应用不仅功能强大，而且在使用上更加符合用户的个性化需求，从而为他们带来更加愉悦和高效的在线体验^[5]。

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>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 XXX 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% ;
```


}

6.3 项目的运行和测试

为了确保基于鼠标的个性化 UI 功能在 Web 应用中稳定且高效地运行，我们需要进行一系列的测试来保障用户体验的质量。首先，编写详细的单元测试，确保每个鼠标事件处理函数和交互逻辑都能准确、无误地响应和执行。这有助于我们在开发早期就发现和修复潜在的问题。其次，进行集成测试，验证不同模块之间的交互是否顺畅，以及整体功能是否如预期般工作。这能够确保系统的各个部分能够协同工作，为用户提供流畅、连贯的交互体验。此外，我们还需要在不同的浏览器环境（如 Chrome、Firefox、Safari、Edge 等）中进行测试，以确保鼠标交互在不同浏览器中的行为保持一致，从而避免因浏览器兼容性问题而影响用户体验。最后，我们需要在各种设备上进行测试，包括不同屏幕尺寸和分辨率的设备，以确保鼠标操作的响应速度和显示效果都能达到最佳状态。这将有助于我们确保应用在各种设备上都能提供优质的用户体验。鼠标模型 PC 端显示如图 5.1 所示，移动端如图 5.2 所示。

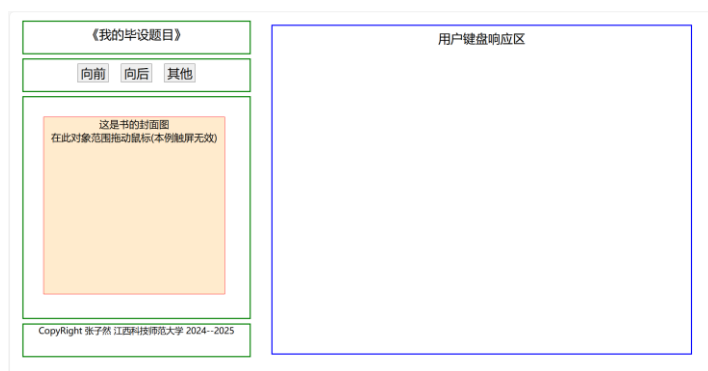


图 6.1 鼠标模型 PC 端

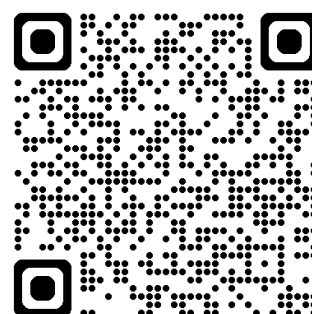


图 6.2 移动端显示

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 1.4.html
```

```
$ git commit -m "1.4.html"
```

成功提交代码后，gitbash 的反馈如下图 6.3 所示：

```
张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)
$ git add 1.4.html

张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)
$ git commit -m "1.4.html"
[master e00f029] 1.4.html
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：

```
张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)
$ git log
commit e00f029f393214ee75d32f0a90cd93ccded8a53f (HEAD -> master)
Author: ZZR229 <2292074611@qq.com>
Date: Tue Jun 18 22:13:18 2024 +0800

    1.4.html

commit 98b588c5b77f496514d8652cc452a08e339b2557
Author: ZZR229 <2292074611@qq.com>
Date: Tue Jun 18 22:01:10 2024 +0800

    1.3.html

commit 6b70c6ec39663d998bb04c5b30513ca3eddf8db4
Author: ZZR229 <2292074611@qq.com>
Date: Tue Jun 18 21:47:38 2024 +0800

    1.2.html
```

图 6.4 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 应用开发中，确保用户体验的一致性和跨设备兼容性是至关重要的。随着用户通过各种设备访问 Web 应用，包括触摸屏移动设备和使用鼠标的桌面电脑，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变得尤为重要。为了实现这一目标，我们需要深入分析用户在不同设备上的交互习惯，精心设计易于理解和操作的界面元素，以及开发能够灵活响应不同输入方式的交互逻辑。以下将详细探讨如何通过分析、设计和实现过程，构建这样一个跨设备、通用且用户友好的交互界面。

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/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 XXX 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%;  
}
```

7.3 项目的运行和测试

在确保 Web 应用的触屏和鼠标通用交互操作能够正常运行并提供优质的用户体验时，我们必须进行全面的测试和优化流程。首先，我们需要编写详细的测试用例，确保各种交互操作在触屏和鼠标两种输入方式下都能如预期般工作。其次，通过集成测试来验证不同交互操作和模块之间的协作是否顺畅，从而确保整体功能的正确性。此外，我们需要在主流的桌面浏览器和移动浏览器中进行广泛的测试，以验证应用在不同环境下的行为一致性和用户体验。同时，我们还需要在各种设备上实际测试应用，包括不同品牌和型号的智能手机、平板电脑以及 PC，以确保在各种屏幕尺寸、分辨率和输入方式下，用户都能获得流畅且一致的体验。为了更深入地了解用户在实际使用中的行为和感受，我们邀请用户在实际设备上应用测试，并观察他们的操作行为。通过问卷调查和用户访谈，我们收集用户对于交互操作的使用体验和改进建议，这些反馈将作为我们进一步优化应用的重要依据。最后，我们关注事件处理的响应速度，通过技术分析和优化，确保无论是触屏还是鼠标操作，应用都能提供迅速且流畅的响应，从而为用户带来愉悦的交互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 7.1PC 端

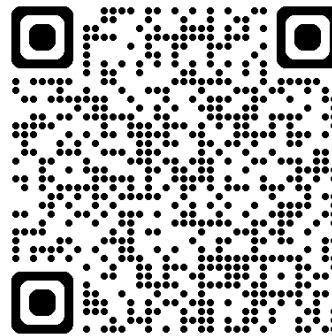


图 7.2 移动端访问

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 1.5.html  
$ git commit -m "1.5.html"
```

成功提交代码后，gitbash 的反馈如下图 7.3 所示：

```
张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)  
$ git add 1.5.html  
  
张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)  
$ git commit -m "1.5.html"  
[master 2e2d680] 1.5.html  
1 file changed, 215 insertions(+)  
create mode 100644 1.5.html
```

图 7.3 gitbash 提交反馈

gitbash 反馈代码的仓库日志如下图 7.4 所示：

```
张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)
$ git log
commit 2e2d680cf245086509ab0f55dd739fba2e889cb0 (HEAD -> master)
Author: ZZR229 <2292074611@qq.com>
Date: Tue Jun 18 22:21:17 2024 +0800

    1.5.html

commit e00f029f393214ee75d32f0a90cd93ccded8a53f
Author: ZZR229 <2292074611@qq.com>
Date: Tue Jun 18 22:13:18 2024 +0800

    1.4.html

commit 98b588c5b77f496514d8652cc452a08e339b2557
Author: ZZR229 <2292074611@qq.com>
Date: Tue Jun 18 22:01:10 2024 +0800

    1.3.html

commit 6b70c6ec39663d998bb04c5b30513ca3eddf8db4
Author: ZZR229 <2292074611@qq.com>
Date: Tue Jun 18 21:47:38 2024 +0800

    1.2.html
```

图 7.4 gitbash 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

在 Web 应用的用户体验中，键盘交互的重要性不容忽视，特别是在桌面环境中。一个精心设计的个性化键盘交互不仅能显著提高用户的操作效率，还能为特定应用场景带来极大的便利。为了满足不同用户的需求和偏好，本节将深入探讨如何设计、实现并有效管理针对键盘的个性化 UI 交互控制。通过细致入微的交互设计，我们旨在为用户带来更加流畅、高效且个性化的键盘操作体验。

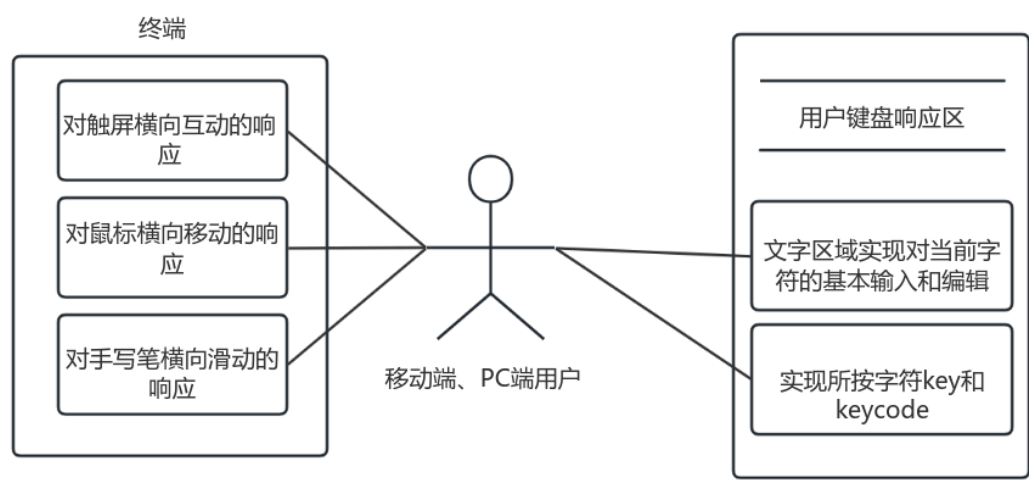


图 Error! No text of specified style in document..2UI 的个性化键盘响应用例图

8.2 项目的实现和编程

二、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">
  本利代码的运行需要超过 1 千像素宽度的宽屏<br>
  建议使用有键盘的 PC 运行和调试代码<br>
  当然拖动/滑动超过 100 像素的 UI 互动依然有效!
</div>

</main>

<footer>
  Copyright XXX 江西科技师范大学 2024--2025
</footer>

<div id="aid">
  用户键盘响应区

  <p id="typeText"></p>
  <p id="keyStatus"></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;
}

footer{
    border: 3px solid green;
    height: 10%;
    font-size: 0.7em;
}

body{
    position: relative;
}

button{
    font-size: 1em;
}

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

```
    left:7% ;  
    top: 7% ;  
}  
#aid{  
    position: absolute;  
    border: 3px solid blue;  
    top:0px;  
    left:600px;  
}  
#typeText{  
    border: 1px solid blue;  
    padding:0.2em ;  
    color: gray ;  
}  
#keyStatus{  
    position: absolute;  
    border: 1px solid blue;  
width: 90% ;  
    right: 0;  
    bottom: 0;  
    font-size: 0.6em;  
    padding:0.5em ;  
}
```

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

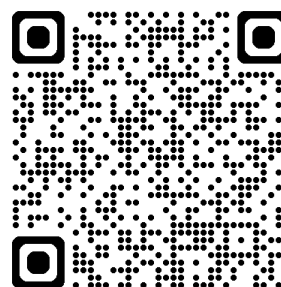


图 8.2 移动端访问

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 1.6.html
```

```
$ git commit -m "1.6.html"
```

成功提交代码后，gitbash 的反馈如下图 8.3 所示：

```
张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)
$ git add 1.6.html

张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)
$ git commit -m "1.6.html"
[master f64cba2] 1.6.html
1 file changed, 255 insertions(+)
create mode 100644 1.6.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：

```
张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)
$ git log
commit f64cba21f26d2411779b67c88f14e4cf96a8dcc1 (HEAD -> master)
Author: ZZR229 <2292074611@qq.com>
Date: Tue Jun 18 23:50:22 2024 +0800

    1.6.html

commit 2e2d680cf245086509ab0f55dd739fba2e889cb0
Author: ZZR229 <2292074611@qq.com>
Date: Tue Jun 18 22:21:17 2024 +0800

    1.5.html

commit e00f029f393214ee75d32f0a90cd93ccded8a53f
Author: ZZR229 <2292074611@qq.com>
Date: Tue Jun 18 22:13:18 2024 +0800

    1.4.html

commit 98b588c5b77f496514d8652cc452a08e339b2557
Author: ZZR229 <2292074611@qq.com>
Date: Tue Jun 18 22:01:10 2024 +0800

    1.3.html

commit 6b70c6ec39663d998bb04c5b30513ca3eddf8db4
Author: ZZR229 <2292074611@qq.com>
Date: Tue Jun 18 21:47:38 2024 +0800

    1.2.html
```

图 8.4 gitbash 日志反馈

9. 谈谈本项目中的高质量代码

在探索 UI 设计的巧妙应用中，键盘控制的个性化设计无疑是一大亮点。尤其是在现代用户界面（UI）中，对键盘底层事件如 `keydown` 和 `keyup` 的精细处理，不仅能够为用户提供更为流畅、便捷的操作体验，还能在某些特定场景下显著提升工作效率。

9.1 响应式设计

尽管文本中提示代码的运行需要超过 1 千像素宽度的宽屏，并建议使用带有键盘的 PC 进行运行和调试，但我们可以进一步探索响应式设计，使界面能够在不同尺寸的屏幕和设备上都能良好地运行和显示。通过媒体查询（Media Queries）和灵活的布局策略，我们可以确保界面在不同设备上都能提供出色的用户体验。

9.2 键盘导航优化

`<nav>` 部分中的按钮“向前”、“向后”和“其他”为我们展示了使用键盘进行页面导航的潜力。我们可以进一步扩展这一功能，通过监听键盘上的特定按键（如箭头键、Tab 键等）来实现页面元素的快速切换和焦点移动。这样的设计能够使用户在不需要鼠标操作的情况下，依然能够流畅地浏览和操作页面。

9.3 键盘输入实时反馈

`#typeText` 元素似乎是一个用于接收用户输入的文本框。我们可以利用 `keydown` 和 `keyup` 事件，实现对用户输入的实时反馈。例如，当用户按下某个键时，我们可以立即在界面上显示该键的键值或对应的操作提示。这样的设计不仅能够增强用户的操作感知，还能在某些需要精确输入的场景下提供有用的指导。

9.4 自定义快捷键

除了基本的键盘导航和输入反馈外，我们还可以为用户提供自定义快捷键的功能。通过允许用户设置自己熟悉的快捷键来触发特定的页面操作或功能，我们可以进一步提高用户的工作效率和使用体验。这样的设计不仅能够满足用户的个性化需求，还能使界面更加灵活和可定制。

9.5 视觉与交互的完美结合

最后但同样重要的是，我们需要确保界面的视觉设计和交互设计能够完美结合。通过精心选择的颜色、字体和布局元素，以及流畅自然的动画效果和交互反馈，我们可以为用户打造一个既美观又易用的界面。这样的设计不仅能够吸引用户的注意力，还能让他们在使用过程中感受到愉悦和满足。

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 GNU 项目开发的强大命令行解释器和脚本语言, 广泛用于 Unix 和类 Unix 系统。自 1989 年发布以来, Bash 已成为系统管理员、开发者和高级用户的首选工具, 用于系统管理和脚本编程。Bash 不仅支持基本的命令执行, 还提供强大的脚本编写功能, 使用户能够自动化日常任务。

Bash 的名字来源于 Bourne shell 的增强版本, 继承了 Bourne shell 的特性并进行了扩展。随着版本的更新, Bash 引入了新的功能和改进, 如数组支持、错误处理和字符串操作。Bash 的兼容性和功能性使其成为 Linux 和 macOS 等系统的默认 shell。

在现代计算环境中, Bash 依然重要。尽管图形用户界面 (GUI) 普及, 但命令行界面 (CLI) 在效率和灵活性上仍不可替代。Bash 特别在 DevOps 和云计算领域受到青睐, 用于配置管理、部署自动化和系统监控等任务。Windows 用户可以通过 GitBash 享受 Bash 的功能, GitBash 集成了 Git 和 Bash, 为 Windows 提供了一个仿 Linux 的环境。

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 江科大张子然
$ git config user.email 2292074611@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'

张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)
$ git add index.html myCSS.css

张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)
$ git commit -m 项目第一版："三段论"式的内容设计概要开发
[master (root-commit) e2eb182] 项目第一版："三段论"式的内容设计概要开发
2 files changed, 150 insertions(+)
create mode 100644 index.html
create mode 100644 myCSS.css

张子然@LAPTOP-OGKK6C54 MINGW64 ~/Des
```

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。

```
张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)
$ git log
commit 2e2d680cf245086509ab0f55dd739fba2e889cb0 (HEAD -> master)
Author: ZZR229 <2292074611@qq.com>
Date: Tue Jun 18 22:21:17 2024 +0800

    1.5.html

commit e00f029f393214ee75d32f0a90cd93ccded8a53f
Author: ZZR229 <2292074611@qq.com>
Date: Tue Jun 18 22:13:18 2024 +0800

    1.4.html

commit 98b588c5b77f496514d8652cc452a08e339b2557
Author: ZZR229 <2292074611@qq.com>
Date: Tue Jun 18 22:01:10 2024 +0800

    1.3.html

commit 6b70c6ec39663d998bb04c5b30513ca3eddf8db4
Author: ZZR229 <2292074611@qq.com>
Date: Tue Jun 18 21:47:38 2024 +0800

    1.2.html
```

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^[6]。

10.3.1 创建空的远程代码仓库

点击主页右上角的+号的下拉菜单的 New repository 进行创建空的远程代码仓库。如图 8.3 所示。

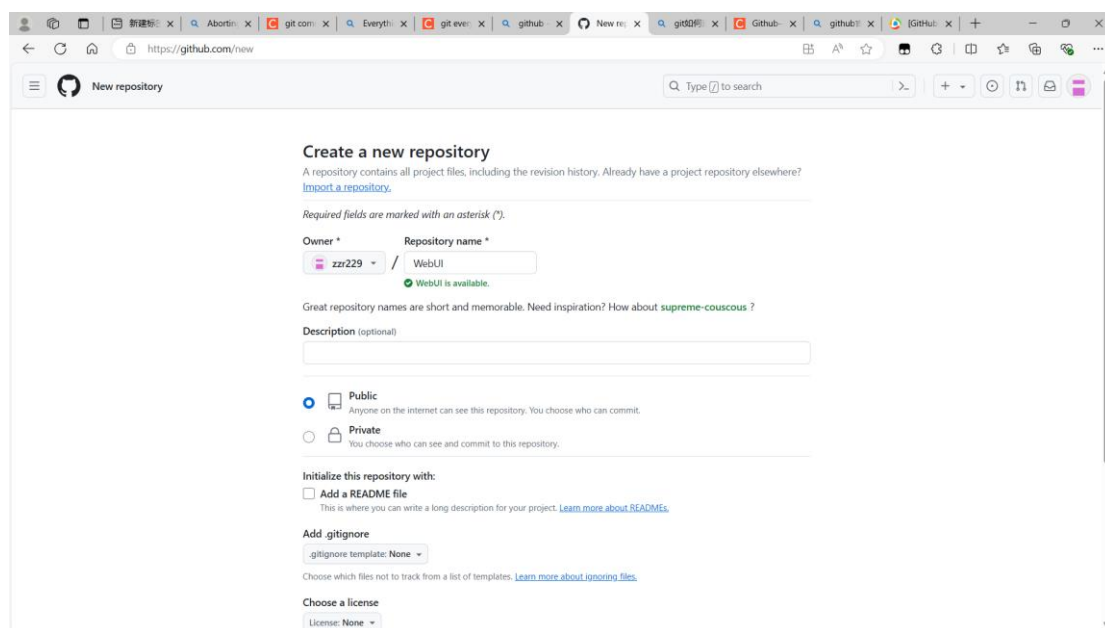


图 8.3 创建空的仓库

10.3.2 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/zxr229/WebUI.git
```

```
张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI
$ git init
Initialized empty Git repository in C:/Users/张子然/Desktop/WebUI/.git/

张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)
$ git git remote add origin https://github.com/zzr229/WebUI.git
git: 'git' is not a git command. See 'git --help'.

The most similar command is
    init

张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)
$ git remote add origin https://github.com/zzr229/WebUI.git

张子然@LAPTOP-OGKK6C54 MINGW64 ~/Desktop/WebUI (master)
$
```

图8.4 本地仓库与远程仓库连接

参考文献

- [1] 张宇薇. HTML5 在 Web 前端开发中的应用[J]. 集成电路应用, 2024,41(04):274-276.
- [2] 许至晶. JavaScript 在 Web 开发的应用研究[J]. 信息记录材料, 2022,23(12):242-244.
- [3] 邹晓丹. 基于 HTML5 和 CSS3 的网页前端设计优化研究[J]. 自动化应用, 2023,64(S1):217-219.
- [4] 金琦. 初识 Git 项目版本管理工具[J]. 中国信息技术教育, 2024(11):83-87.
- [5] 吴倩. UI 界面设计中视觉要素的作用分析与应用研究[J]. 天工, 2023(20):48-50.
- [6] 张海清, 赵卫东, 李代伟, 等. 基于 Github 的软件工程实验教学平台的研究[J]. 创新创业理论研究与实践, 2022,5(20):159-161.