

成绩:

# 江西科技师范大学

## 课程设计（论文）

题目（中文）：基于 Web 客户端技术的个性化 UI 设计和实现

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

院（系）：元宇宙产业学院

专    业：计算机科学与技术

学生姓名：钟雅婷

学    号：20213637

指导教师：李健宏

2024 年 6 月 18 日

# 目录

1. 前言.....	2
1.1 毕设任务分析.....	2
1.2 研学计划.....	2
1.3 研究方法.....	3
2. 技术总结和文献综述.....	4
2.1 Web 平台和客户端技术概述.....	4
2.2 项目的增量式迭代开发.....	4
3. 内容设计概要.....	5
3.1 分析和设计.....	5
3.2 项目的实现和编程.....	6
3.3 项目的运行和测试.....	7
3.4 项目的代码提交和版本管理.....	7
4. 移动互联时代的 UI 开发初步—窄屏终端的响应式设计.....	8
4.1 分析和设计.....	8
4.2 项目的实现和编程.....	9
4.3 项目的运行和测试.....	10
4.4 项目的代码提交和版本管理.....	11
5. 应用响应式设计技术开发可适配窄屏和宽屏的 UI.....	12
5.1 分析和设计.....	12
5.2 项目的实现和编程.....	13
5.3 项目的运行和测试.....	14
5.4 项目的代码提交和版本管理.....	15
6. 个性化 UI 设计中对鼠标交互的设计开发.....	15
6.1 分析和设计.....	15
6.2 项目的实现和编程.....	16
6.3 项目的运行和测试.....	18
6.4 项目的代码提交和版本管理.....	19
7. 对触屏和鼠标的通用交互操作的设计开发.....	19
7.1 分析和设计.....	19
7.2 项目的实现和编程.....	20
7.3 项目的运行和测试.....	23
7.4 项目的代码提交和版本管理.....	23
8. UI 的个性化键盘交互控制的设计开发.....	24
8.1 分析和设计.....	24
8.2 项目的实现和编程.....	25
8.3 项目的运行和测试.....	28
8.4 项目的代码提交和版本管理.....	28
9. 用 gitBash 工具管理项目的代码仓库和 http 服务器.....	29
9.1 经典 Bash 工具介绍.....	29
9.2 通过 GitHub 平台实现本项目的全球域名.....	29
9.3 创建一个空的远程代码仓库.....	30
9.4 设置本地仓库和远程代码仓库的链接.....	30

参考文献.....	33
-----------	----

# 基于 Web 客户端技术的个性化 UI 设计和实现

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

科师大元宇宙产业学院 2021 级 钟雅婷

**摘要:**在过去十年中,HTML5 作为 Web 标准的核心,以其跨平台和开源特性,在软件开发领域取得了广泛应用。本项目选择以 HTML5 为基础的 Web 客户端技术,进行程序设计和软件开发的深入研究与实践。本项目深入探索了 HTML 内容建模、CSS 样式设计和 JavaScript 功能编程的基本技术和技巧。并为了实现一个个性化的用户界面(UI),采用了响应式设计,能够智能适配不同屏幕,满足 PC 端和移动端设备的多样化需求。在功能上,项目利用 DOM 技术和事件驱动模式,实现了对鼠标、触屏、键盘的底层事件响应和流畅支持。项目创新性地设计了一个通用的 pointer 模型,用一套代码实现了对鼠标和触屏的控制,提升了代码的质量和复用性。在工程管理方面,项目采用了增量式开发模式,通过六次迭代开发,每次迭代都严格遵循了分析、设计、实现、测试四个经典开发阶段,逐步优化和完善 UI 应用程序,确保了项目设计的高效性和开发过程的可控性。为了促进代码的开源和分享,项目使用了 Git 工具进行版本管理,详细记录了开发过程和代码优化的每一步。在开发过程中,共进行了六次代码重构。最后通过 Gitbash 工具,项目将代码仓库上传到 GitHub,并利用 GitHub 提供的 HTTP 服务器,实现了 UI 应用的全球部署。用户现在可以通过一个简单的地址或二维码,便捷地跨平台访问这个程序,体现了项目的创新性、实用性和开放性。

**关键词:** Web 客户端技术; 响应式设计; 事件驱动模式; 版本管理与开源

# 1. 前言

## 1.1 毕设任务分析

毕业设计是指在高等学府学习期间，根据所学专业知识和技能，结合实际问题或主题，进行设计、实施和评估研究项目的过程。毕业设计旨在让学生通过实践应用所学知识，提高解决问题的能力，并培养创新思维和专业技能。毕业设计通常是一个实践性较强的项目，学生需要展示对所学知识的深入理解和运用能力。

毕业论文则是基于对某一特定课题或问题的研究，撰写出具有一定学术价值和贡献的书面论述。毕业论文要求学生具备独立思考、分析问题和撰写文献的能力，结合相关理论和实证研究，对问题进行深入探讨和解答。毕业论文不仅要展现学生的学术水平，还要体现其逻辑思维、批判性思考和表达能力<sup>[1]</sup>。

## 1.2 研学计划

研究计划是确保项目按时完成的蓝图。它应该包括一个清晰的项目目标陈述，这些目标将指导整个设计和研究过程。计划中应详细说明每个阶段的预期成果，包括文献综述、技术选型、原型开发、用户测试和最终的论文撰写。同时，计划还应包括时间表，明确每个阶段的开始和结束时间，以及关键的里程碑<sup>[2]</sup>。此外，资源分配也是计划的一部分，需要考虑所需的技术工具、软件、硬件以及可能的人力资源。风险评估同样重要，它帮助识别可能的障碍，并提前规划应对策略。我的毕业论文进度表如下所示。

毕业论文（设计）进度计划			
序号	各阶段工作内容	起讫日期	备注
一	定选题	2024 年 10 月 7 日—10 月 10 日	指导教师向学生下达任务书
二	开题答辩	2024 年 10 月 14 日星期六、10 月 15 日星期日	进行 2025 届 2021 级毕业论文开题答辩，制作开题报告答辩PPT，答辩时间 10 分钟（含提问时间）。
三	文献综述	2024 年 10 月 16 日—10 月 31 日	完成文献综述的撰写，文献综述要求 3000 字。
四	论文一稿	2024 年 11 月 1 日—11 月 30 日	先撰写论文框架

五	论文二稿	2024 年 12 月 1 日—12 月 31 日	再撰写论文内容
六	中期检查	2025 年 2 月 1 日—2 月 7 日	项目作品完成，论文完成一半的进度。
七	论文三稿	2025 年 1 月 4 日—1 月 31 日	论文格式按照学术规范
八	论文定稿	2025 年 2 月 26 日—4 月 12 日	做好中国知网学术不端审核检测自查工作
九	毕业论文（设计）中国知网学术不端审核检测自查工作	2025 年 4 月 12 日-14 日	学生提交中国知网重复率检测报告，落实本科毕业生毕业论文（设计）中国知网学术不端审核检测自查工作，文件名称：学号-项目班级-姓名，汇总提交给指导老师。
十	本科毕业生毕业论文（设计）中国知网学术不端审核检测	2025 年 4 月 15 日—4 月 19 日	进行本科毕业生毕业论文（设计）中国知网学术不端审核检测，提交中国知网重复率自查达标的论文Word文档，重复率 $\leq 15\%$ 。毕业论文文件名：学号-姓名-行政班级-论文名称-指导老师.docx，按行政班汇总。

### 1.3 研究方法

研究方法涉及具体的技术路线和实施步骤。首先，通过广泛的文献综述来建立研究的理论基础，这包括对现有研究的深入分析，以及对个性化 UI 设计和 Web 客户端技术的最新进展的了解。接着，采用系统化的技术学习方法，可能包括在线课程、研讨会或自学，以掌握所需的前端开发技能。设计方法论上，采用用户中心设计，确保设计方案紧密贴合用户需求。在原型开发阶段，可以采用敏捷开发方法，通过快速迭代来不断改进设计。用户测试是评估设计的关键环节，通过可用性测试和 A/B 测试等方法收集用户反馈。数据分析方法将用于解读测试结果，以指导设计的优化。最后，论文撰写将遵循学术规范，确保研究的系统性和严谨性。代码和文档管理是研究方法的一部分，通过版本控制系统来维护项目文档和代码的一致性和可追溯性。

## 2. 技术总结和文献综述

### 2.1 Web 平台和客户端技术概述

从最初的静态网页到动态内容的交互，Web 平台经历了巨大的变革。Tim Berners-Lee 发明了 Web 的基本技术架构，并成立了 W3C 组织，致力于推动 Web 标准的发展。HTML5 和 ECMAScript 的推出标志着 Web 技术的一个重要里程碑，它们为开发者提供了统一的开发平台，使得 Web 应用能够跨平台运行。

随着技术的进步，Web 平台已经从简单的文档分享演变为一个功能丰富的应用运行环境。现代 Web 平台支持多媒体内容、交互式应用、网络服务集成等，为用户提供了丰富的在线体验<sup>[3]</sup>。

### 2.2 项目的增量式迭代开发

软件工程视角中，软件生命周期中主要包括开发和维护两项工作，其中在我们的本科毕业设计中仅涉及开发过程。软件的开发过程包括四个阶段：分析 (analysis)、设计 (design)、实现 (implementation) 和测试 (testing)，下文简称为 ADIT。而开发过程，有经典的两种模型：瀑布模型和增量模型。

在瀑布模型中，开发过程是非常严格且单向的。在 ADIT 的四个阶段流程中，只有在上一个阶段完成之后才能启动下一个阶段。比如，我们在实施过程中发现设计过程中的问题，是无法推翻设计过程的。这一点对普通开发者和小团队是非常不方便的。因此我的结课设计采用了增量开发模式。

在增量式的软件开发模型中，开发者分一系列步骤进行开发。我们首先完成了整个系统的一个简化版本，这个版本表示整个系统，但不包括系统内部的详细设计。图 2-1 表达了增量式的软件开发模型的概念。在后续版本的开发中，通过分析和设计添加了更多的系统软件的细节，然后再次进行系统部署的开发和测试。若在编程和测试中发现了设计甚至是分析中的问题，开发人员就能够及时反馈以及更改设计<sup>[4]</sup>。

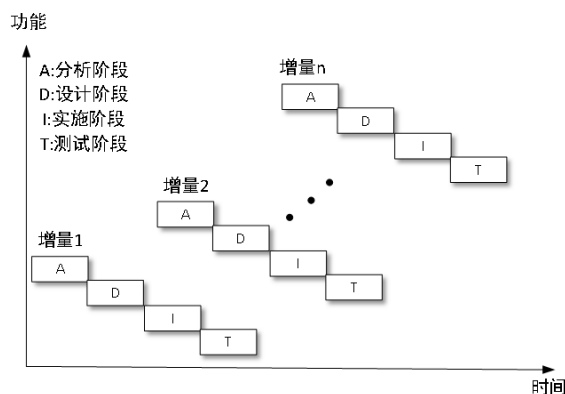


图 2-1 增量模型的概念

在增量模型中，软件分一系列步骤进行开发。开发人员首先完成了整个系统的一个简化版本。这个版本表示整个系统，但不包括详细信息。图中显示了增量模型的概念。

在第二个版本中，添加了更多的细节，而一些没有完成，系统再次测试。如果有问题，开发人员就知道问题在于新功能。在现有的系统正常工作之前，它们不会添加更多的功能。此过程，直到添加所有所需的功能<sup>[5]</sup>。

### 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，本项目使用“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 用例图所示，完整的 DOM 树如图 3-2 所示。

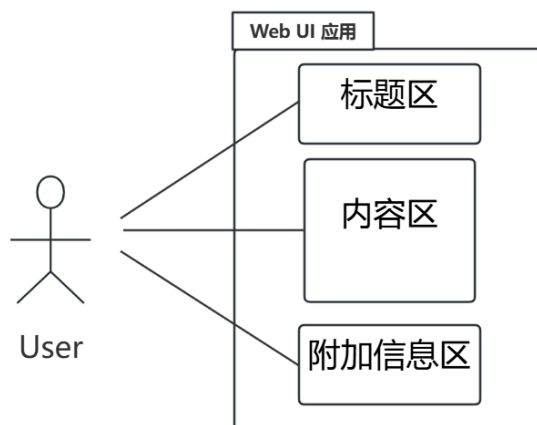


图 3-1 内容设计概要的用例图



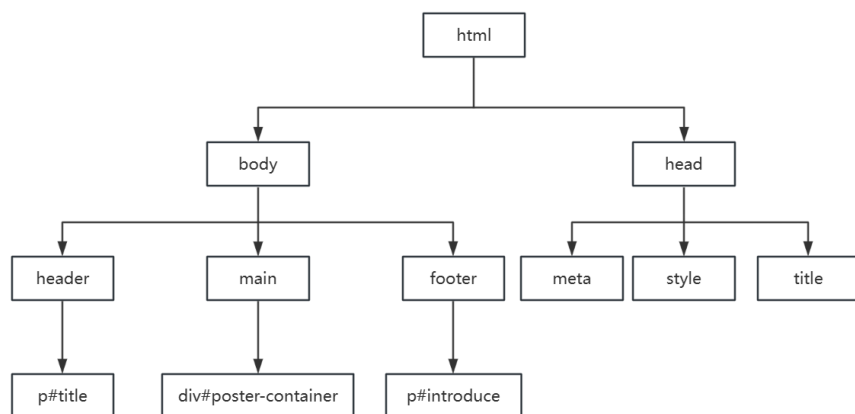


图 3-2 内容设计概要的 DOM 树

## 3.2 项目的实现和编程

(1)HTML 代码编写如下：

```

<header>
    《 基于 SSM 的健身房会员管理系统的设计与实现 》
</header>
<main>
    （内容区）
</main>
<footer>
    Copyright 钟雅婷 江西科技师范大学 2024-2025
</footer>
  
```

(2)CSS 代码编写如下：

```

*{
    margin: 10px;
font-size:25px ;
}
header{
    border: 2px solid #B0C4DE;
    height: 100px;
text-align: center;
}
main{
    border: 2px solid #B0C4DE;
    height: 400px; }
footer{
    border: 2px solid #B0C4DE;
    height: 100px;
text-align: center;
}
  
```

### 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 Edge 浏览器打开项目的结果，如下图 3-3 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 3-4 的二维码，运行测试本项目的第一次开发的阶段性效果。网址为：<https://zyt20020915.github.io/1.1.html>



图 3-3 PC 端运行效果图

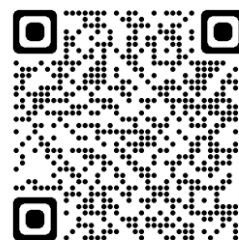


图 3-4 移动端二维码

### 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 /d
$ mkdir webUI
$ cd webUI
$ git init
$ git config user.name 钟雅婷
$ git config user.email 3261458337@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 的反馈如下所示：

```
32614@LAPTOP-JSNMSI8H MINGW64 /d/webUI (master)
$ git commit -m 项目第一版: "三段论"式的内容设计概要开发
[master ae6a89d] 项目第一版: 三段论式的内容设计概要开发
18 files changed, 45 insertions(+)
create mode 100644 mycss1.css
rename index.html => mycss2.css (100%)
```

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所示：

```
32614@LAPTOP-JSNMSI8H MINGW64 /d/webUI (master)
$ git log
commit ae6a89dd78fb56d08c37017953e6e9709ec57ad7 (HEAD -> master)
Author: 钟雅婷 <3261458337@qq.com>
Date: Mon Jun 3 19:53:28 2024 +0800
```

项目第一版：三段论式的内容设计概要开发

## 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

这一步是项目开发的第二阶段，响应式设计是一种让网站能够适应不同设备屏幕尺寸和分辨率的设计方法。它使得网站能够在手机、平板、桌面电脑等不同设备上提供良好的用户体验。本项目通过把 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，并且通过 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。如下图 4-1 所示，是不同手机用户的用例图。如图 4-2 是该项目完整的 DOM 树。

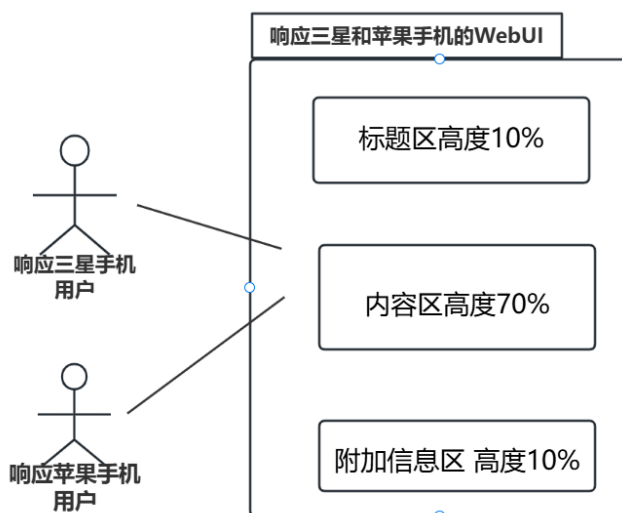


图 4-1 不同手机用户的用例图

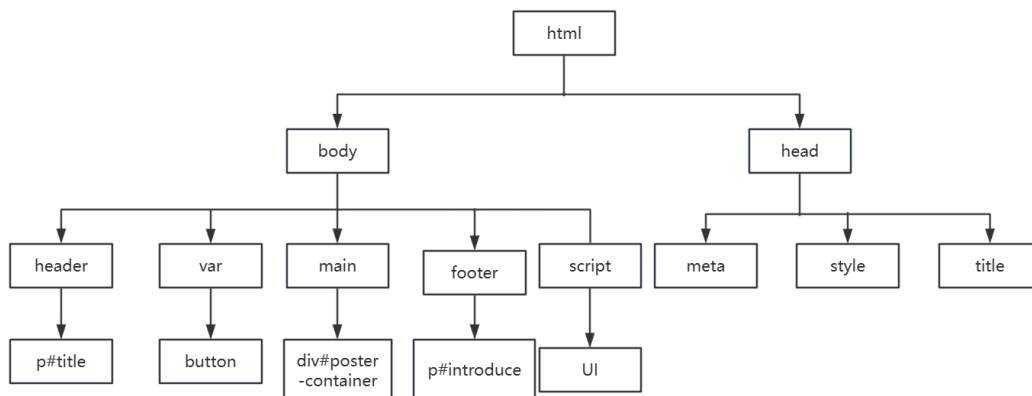


图 4-2 窄屏终端的响应式设计的 DOM 树

## 4.2 项目的实现和编程

(1) HTML 代码编写如下

```

<header>
  <p id="book">
    《 基于 SSM 的健身房会员管理系统的设计与实现 》
  </p>
</header>
<nav>
  <button>导航一</button>
  <button>导航二</button>
  <button>导航三</button>
</nav>
<main id = 'main'>
  健身房会员管理系统主要分为会员管理、充值管理、教练管理、课程管理、器材管理、物品
  遗失管理、商品管理、信息统计管理、系统管理等这几个方面。
</main>
<footer>
  <p id="statusInfo">
    钟雅婷 @ 江西科技师范大学 2025
  </p>
</footer>

```

(2) 与上一阶段比较，本阶段初次引入了 em 和 % ，这是 CSS 语言中比较高阶的语法，可以有效地实现我们的响应式设计，CSS 部分代码编写如下。

```

*{
  margin: 10px;
  text-align: center;
}
header{
  border: 2px solid #B0C4DE;
}

```

```

    height: 15%;
    font-size: 1em;

}
main{
    border: 2px solid #B0C4DE;
    height: 70%;
    font-size: 1.2em;

}
nav{
    border: 2px solid #B0C4DE;
    height: 10%;
}
nav button{
    font-size: 1.1em;
}
footer{
    border: 2px solid #B0C4DE;
    height: 7%;
}

```

(3) 与上一阶段比较，本阶段首次使用了 JavaScript，首先创建了一个 UI 对象，然后把系统的宽度和高度记录在 UI 对象中，又计算了默认字体的大小，最后再利用动态 CSS，实现了软件界面的全屏设置。JS 部分代码编写如下。

```

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 - 2*baseFont + "px" ;
document.body.style.height = UI.appHeight - 4*baseFont + "px";

```

### 4.3 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，本文此处给出三星手机和苹果手机端用户的测试图，如下图 4-3 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 4-4 的二维码，运行测试本项目的第二次开发的阶段性效果。网址为：<https://zyt20020915.github.io/1.2.html>



图 4-3 三星手机（左）和苹果手机（右）端用户测试图



图 4-4 移动端二维码

#### 4.4 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，作为项目的第二次迭代，在代码提交和版本管理环节，进入 gitBash 命令行后，按次序输入以下命令：

```
$ cd /d
$ cd webUI
$ git init
```

编写好 phase2.html、mycss2.css 和 myjs2.js 的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add phase/phase2.html mycss2.css mgjs2.js
$ git commit -m 项目第二版：移动互联时代的响应式设计（窄屏）
```

成功提交代码后，gitbash 的反馈如下所示：

```
32614@LAPTOP-JSNMSI8H MINGW64 /d/webUI (master)
$ git commit -m 项目第二版：移动互联时代的响应式设计（窄屏）
[master 06940a7] 项目第二版：移动互联时代的响应式设计（窄屏）
3 files changed, 73 insertions(+)
```

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所示：

```
32614@LAPTOP-JSNMSI8H MINGW64 /d/webUI (master)
$ git log
commit 06940a76daa553d151109950d8210c2d25ec3b03 (HEAD -> master)
Author: 钟雅婷 <3261458337@qq.com>
Date: Mon Jun 3 21:13:57 2024 +0800

    项目第二版：移动互联时代的响应式设计（窄屏）
```

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

这一步是项目开发的第三阶段，本项目增加 user-scalable=no，可以让移动设备的屏幕宽度和高度，与 window 对象的 innerWidth 和 innerHeight 精确对应。在 JS 里尝试对鼠标设计 UI 控制，鼠标在该区域内可以显现该位置的坐标。该项目用例图如图 5-1 所示。该项目完整的 DOM 树如图 5-2 所示。

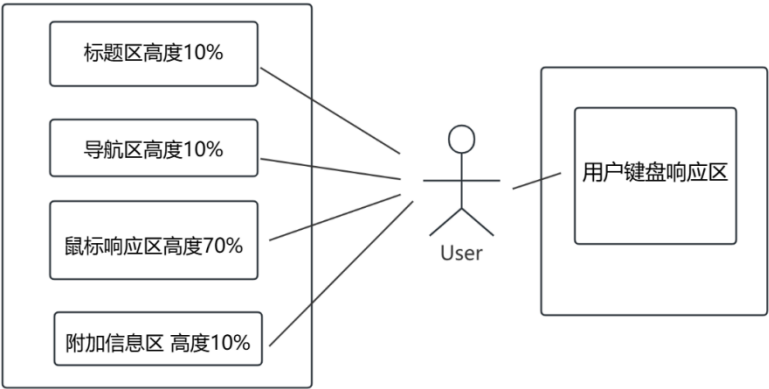


图 5-1 应用响应式设计技术开发可适配窄屏和宽屏的 UI 的用例图

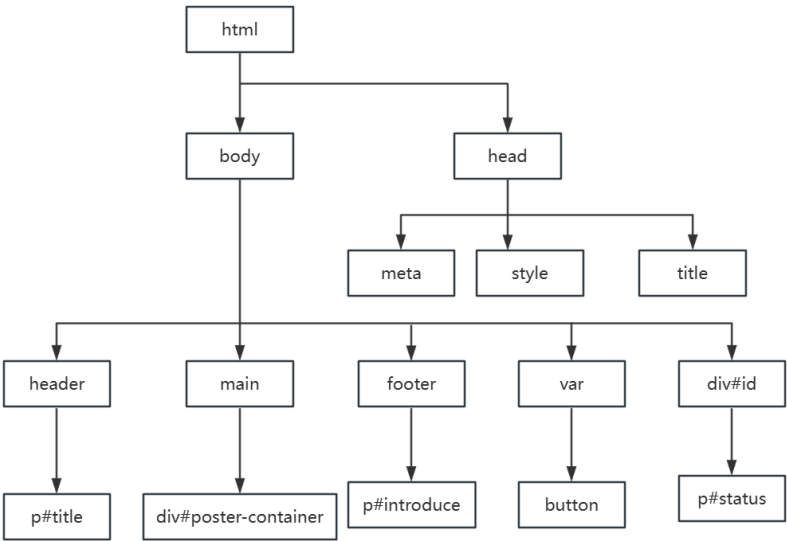


图 5-2 应用响应式设计技术开发可适配窄屏和宽屏的 UI 的 DOM 树

## 5.2 项目的实现和编程

(1) CSS 部分代码编写如下

```
*{
    text-align: center;
    box-sizing: border-box ;
}
header,main,div#bookface,nav,footer{
    margin:1em;
}
header{
    border: 2px solid #B0C4DE;
    height: 20%;
    font-size: 1em;
}
main{
    border: 2px solid #B0C4DE;
    height: 70%;
    font-size: 1.2em;
}
nav{
    border: 2px solid #B0C4DE;
    height: 10%;
}
nav button{
    font-size: 1.1em;
}
footer{
    border: 2px solid #B0C4DE;
    height: 15%;
}
body{
    position:relative ;
}
#aid{
    position: absolute;
    border: 3px solid #B0C4DE;
    top: 0.5em;
    left: 600px;
}
#bookface{
    width: 80%;
    height: 80%;
    border:1px solid red;
```



```
background-color: blanchedalmond;
margin:auto;
}
```

(2) JS 部分代码编写如下

```
//尝试对鼠标设计 UI 控制
var mouse={};
mouse.isDown= false;
mouse.x= 0;
mouse.deltaX=0;
$("#bookface").addEventListener("mousedown",function(ev){
    let x= ev.pageX;
    let y= ev.pageY;
    console.log("鼠标按下了，坐标为: "+"("+x+", "+y+")");
    $("#bookface").textContent= "鼠标按下了，坐标为: "+"("+x+", "+y+")";
});
$("#bookface").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+")");
    $("#bookface").textContent= "鼠标正在移动，坐标为: "+"("+x+", "+y+")";
});
$("#bookface").addEventListener("mouseout",function(ev){
    //console.log(ev);
    $("#bookface").textContent="鼠标已经离开";
});
```

### 5.3 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 Edge 浏览器打开项目的结果，如下图 5-3 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 5-4 的二维码，运行测试本项目的第三次开发的阶段性效果。网址为：<https://zyt20020915.github.io/1.3.html>

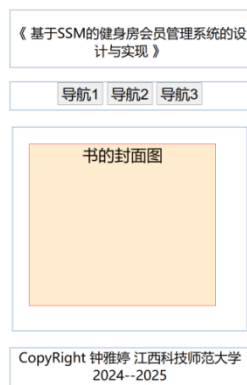


图 5-3 PC 端运行效果图



图 5-4 移动端二维码

## 5.4 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，作为项目的第三次迭代，在代码提交和版本管理环节，进入 gitBash 命令行后，按次序输入以下命令：

```
$ cd /d
$ cd webUI
$ git init
```

编写好 phase3.html、mycss3.css 和 myjs3.js 的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add phase/phase3.html mycss3.css mgjs3.js
$ git commit -m 项目第三版：响应式设计（宽屏和窄屏）
```

成功提交代码后，gitbash 的反馈如下所示：

```
32614@LAPTOP-JSNMSI8H MINGW64 /d/webUI (master)
$ git commit -m 项目第三版：响应式设计（宽屏和窄屏）
[master bf533d4] 项目第三版：响应式设计（宽屏和窄屏）
3 files changed, 151 insertions(+)
```

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所示：

```
32614@LAPTOP-JSNMSI8H MINGW64 /d/webUI (master)
$ git log
commit bf533d4cef875d01e4dfe3cc0a92b4925bd4aa01 (HEAD -> master)
Author: 钟雅婷 <3261458337@qq.com>
Date: Mon Jun 3 21:37:30 2024 +0800
```

项目第三版：响应式设计（宽屏和窄屏）

## 6. 个性化 UI 设计中对鼠标交互的设计开发

### 6.1 分析和设计

这一步是项目开发的第四阶段，本阶段在第三阶段的基础上，实现了鼠标可以横向拖动方块，鼠标点击时也可以显现鼠标坐标，尝试对鼠标的 UI 控制成功。该项目的用例图如图 6-1 所示。该项目的完整 DOM 树如图 6-2 所示。

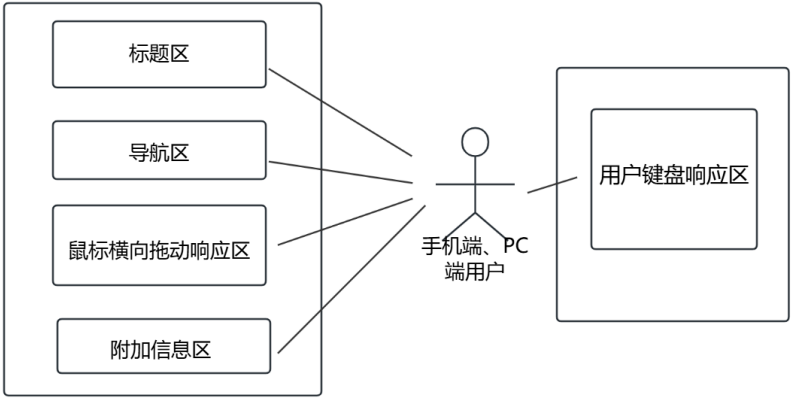


图 6-1 个性化 UI 设计中对鼠标的设计开发的用例图

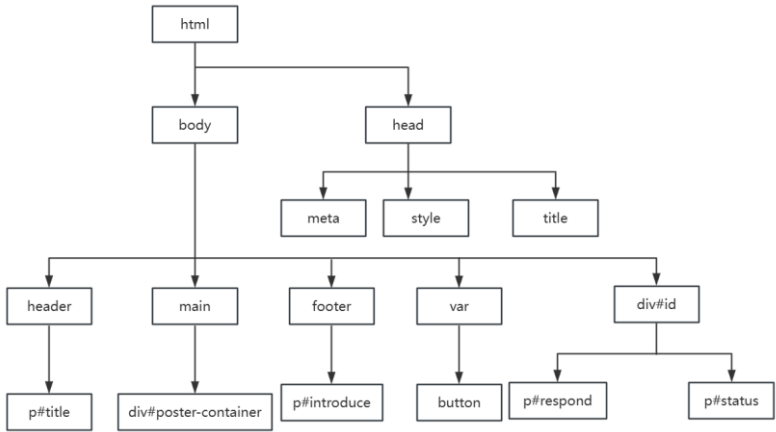


图 6-2 个性化 UI 设计中对鼠标的设计开发的 DOM 树

## 6.2 项目的实现和编程

(1) CSS 代码编写如下

```
*{
    margin: 10px;
    text-align: center;
}
header{
    border: 3px solid #B0C4DE;
    height: 15%;
    font-size: 1em;
}
nav{
    border: 3px solid #B0C4DE;
```

```

    height: 10%;
}
main{
    border: 3px solid #B0C4DE;
    height: 70%;
    font-size: 0.8em;
    position: relative;
}

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

footer{
    border: 3px solid #B0C4DE;
    height:10%;
    font-size: 0.7em;
}
body{
    position: relative;
}
button{
    font-size:1em;
}
#aid{
    position: absolute;
    border: 3px solid #B0C4DE;
    top:0px;
    left:600px;
}
#bookface{
    position: absolute;
    width: 80%;
    height: 80%;
    border:1px solid #B0C4DE;
    background-color: blanchedalmond;
    left:7% ;
    top: 7% ;
}

```

(2) JS 部分代码编写如下

```

$("bookface").addEventListener("mouseup",function(ev){

```

```

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

    $("#bookface").textContent= "鼠标松开!";
    if(Math.abs(mouse.deltaX) > 100){
        $("#bookface").textContent += " 这次是有效拖动! " ;
    }else{
        $("#bookface").textContent += " 本次算无效拖动! " ;
        $("#bookface").style.left = '7%' ;
    }
});
$("#bookface").addEventListener("mousemove",function(ev){
    ev.preventDefault();
    if (mouse.isDown){
        console.log("mouse isDown and moving");
        mouse.deltaX = parseInt( ev.pageX - mouse.x );
        $("#bookface").textContent= "正在拖动鼠标, 距离: " + mouse.deltaX+"px 。";
        $('#bookface').style.left = mouse.deltaX + 'px' ;
    }
});

```

## 6.3 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 Edge 浏览器打开项目的结果, 如下图 6-3 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 6-4 的二维码, 运行测试本项目的第四次开发的阶段性效果。网址为: <https://zyt20020915.github.io/1.4.html>

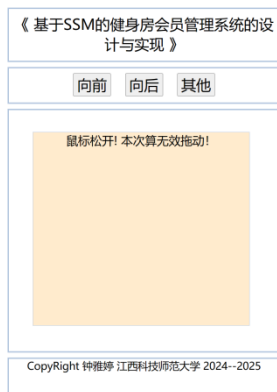


图 6-4 移动端二维码

## 6.4 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，作为项目的第四次迭代，在代码提交和版本管理环节，进入 gitBash 命令行后，按次序输入以下命令：

```
$ cd /d
$ cd webUI
$ git init
```

编写好 phase4.html、mycss4.css 和 myjs4.js 的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add phase/phase4.html mycss4.css myjs4.js
$ git commit -m 项目第四版：个性化 UI 设计中对鼠标交互的设计开发
```

成功提交代码后，gitbash 的反馈如下所示：

```
32614@LAPTOP-JSNMSI8H MINGW64 /d/webUI (master)
$ git commit -m 项目第四版：个性化UI设计中对鼠标交互的设计开发
[master 5874f3c] 项目第四版：个性化UI设计中对鼠标交互的设计开发
3 files changed, 182 insertions(+)
```

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所示：

```
32614@LAPTOP-JSNMSI8H MINGW64 /d/webUI (master)
$ git log
commit 5874f3c2fab079456f1907d7d10e8cdd08ec34ce (HEAD -> master)
Author: 钟雅婷 <3261458337@qq.com>
Date: Thu Jun 6 19:16:52 2024 +0800
```

项目第四版：个性化UI设计中对鼠标交互的设计开发

## 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

这一步是项目开发的第五阶段，在前面四个阶段的基础上，添加了一个用户键盘响应区，在文字输入区域内可以显现所按字符，该项目的用例图如图 7-1 所示。完整的 DOM 树如图 7-2 所示。

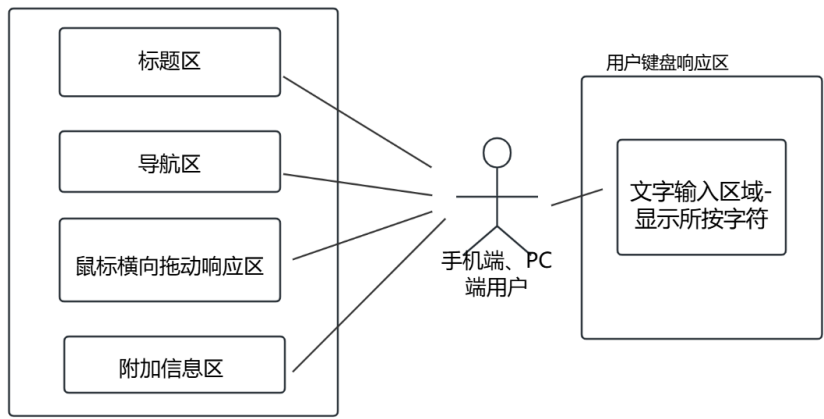


图 7-1 对触屏和鼠标的通用操作的设计开发的用例图

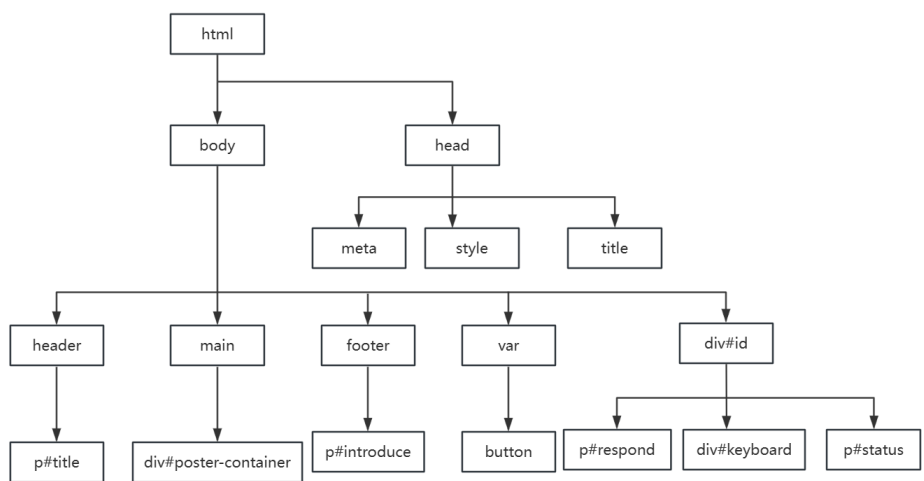


图 7-2 对触屏和鼠标的通用操作的设计开发的 DOM 树

## 7.2 项目的实现和编程

(1) HTML 代码编写如下

```
<header>
  <p id="book">
    《 基于 SSM 的健身房会员管理系统的设计与实现 》
  </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 钟雅婷 江西科技师范大学 2024--2025
</footer>
<div id="aid">
    <p>用户键盘响应区</p>
</div>

```

(2) CSS 代码编写如下

```

*{
    margin: 10px;
    text-align: center;
}
header{
    border: 3px solid #B0C4DE;
    height: 15%;
    font-size: 1em;
}
nav{
    border: 3px solid #B0C4DE;
    height: 10%;
}
main{
    border: 3px solid #B0C4DE;
    height: 70%;
    font-size: 0.8em;
    position: relative;
}
#box{
    position: absolute;
    right: 0;
    width: 100px;
}
footer{

```



```

    border: 3px solid #B0C4DE;
    height:10%;
    font-size: 0.7em;
}
body{
    position: relative;
}
button{
    font-size:1em;
}
#aid{
    position: absolute;
    border: 3px solid #B0C4DE;
    top:0px;
    left:600px;
}
#bookface{
    position: absolute;
    width: 80%;
    height: 80%;
    border:1px solid #B0C4DE;
    background-color: blanchedalmond;
    left:7% ;
    top: 7% ;
}

```

(3) JS 部分代码编写如下

```

$("bookface").addEventListener("mousedown",handleBegin );
$("bookface").addEventListener("touchstart",handleBegin );
$("bookface").addEventListener("mouseup", handleEnd );
$("bookface").addEventListener("touchend",handleEnd );
$("bookface").addEventListener("mouseout", handleEnd );
$("bookface").addEventListener("mousemove", handleMoving);
$("bookface").addEventListener("touchmove", handleMoving);
$("body").addEventListener("keypress", function(ev){
    $("aid").textContent += ev.key ;
});
} //Code Block end
function $(ele){
    if (typeof ele !== 'string'){
        throw("自定义的$函数参数的数据类型错误，实参必须是字符串！");
        return
    }
    let dom = document.getElementById(ele) ;

```

```

        if(dom){
            return dom ;
        }else{
            dom = document.querySelector(ele) ;
            if (dom) {
                return dom ;
            }else{
                throw("执行$函数未能在页面上获取任何元素，请自查问题！");
                return ;
            }
        }
    }
} //end of $

```

### 7.3 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 Edge 浏览器打开项目的结果，如下图 7-3 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 7-4 的二维码，运行测试本项目的第五次开发的阶段性效果。网址为：<https://zyt20020915.github.io/l.5.html>

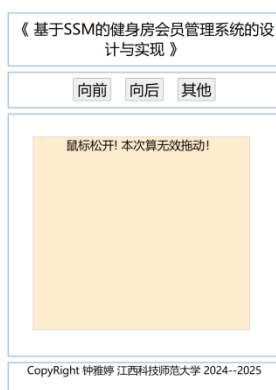


图 7-3 PC 端运行效果图

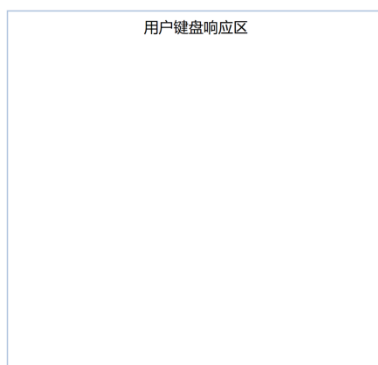


图 7-4 移动端二维码

### 7.4 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，作为项目的第五次迭代，在代码提交和版本管理环节，进入 gitBash 命令行后，按次序输入以下命令：

```

$ cd /d
$ cd webUI
$ git init

```

编写好 phase5.html、mycss5.css 和 myjs5.js 的代码，测试运行成功后，执行下面命令提交代码：

```

$ git add phase/phase5.html mycss5.css mgjs5.js
$ git commit -m 项目第五版：对触屏和鼠标的通用交互操作的设计开发

```

成功提交代码后，gitbash 的反馈如下所示：

```
32614@LAPTOP-JSNMSI8H MINGW64 /d/webUI (master)
$ git commit -m 项目第五版：对触屏和鼠标的通用交互操作的设计开发
[master 01717cf] 项目第五版：对触屏和鼠标的通用交互操作的设计开发
3 files changed, 208 insertions(+)
```

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所示：

```
32614@LAPTOP-JSNMSI8H MINGW64 /d/webUI (master)
$ git log
commit 01717cf5440c7e7703a5d6328255010f879f187e (HEAD -> master)
Author: 江科师大钟雅婷 <3261458337@qq.com>
Date: Thu Jun 6 20:03:47 2024 +0800
```

## 8. UI 的个性化键盘交互控制的设计开发

### 8.1 分析和设计

这一步是项目开发的第六阶段，也是最后一个阶段。把可横向移动的方块里传入照片，并用导航区的按钮进行交互，可以向前向后切换照片，方块同样可以横向移动；同时在用户键盘响应区域增加一个可以显示所按字符 key 值和 keyCode 值。该项目的用例图如图 8-1 所示。UI 的个性化键盘控制的设计开发的完整 DOM 树如图 8-2 所示。

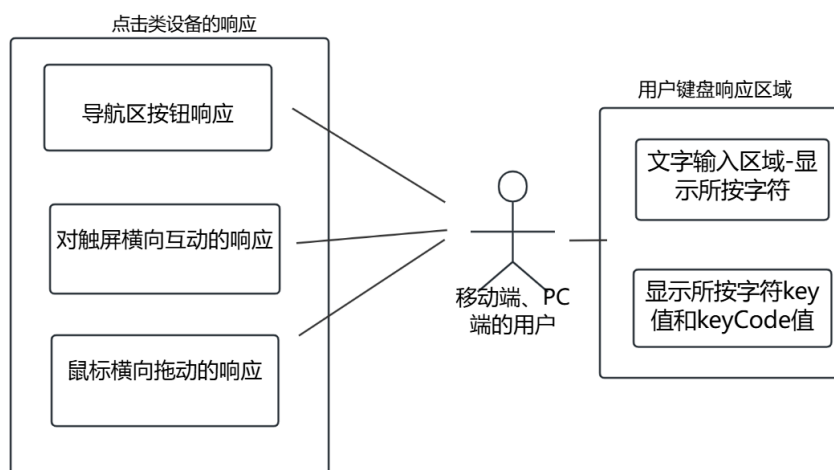


图 8-1 UI 的个性化键盘控制的设计开发的用例图

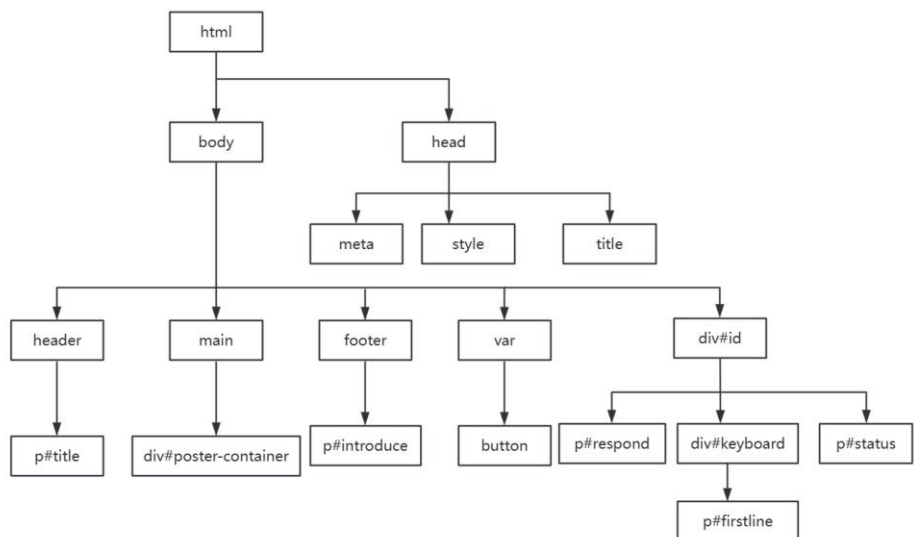


图 8-2 UI 的个性化键盘控制的设计开发的完整 DOM 树

## 8.2 项目的实现和编程

(1) HTML 代码编写如下

```

<header>
  <p id="book">
    小涵同学
  </p>
</header>
<nav>
  <button id="prev">向前</button>
  <button id="next">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 钟雅婷 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>
  
```

(2) CSS 代码编写如下

```
*{
    margin: 10px;
    text-align: center;
    font-family:幼圆;
    font-style: oblique ;
    color:white ;
}
header{
    border: 3px solid #B0C4DE;
    height: 10%;
    font-size: 1em;
    background-color: #B0C4DE;
}
nav{
    border: 3px solid #B0C4DE;
    height: 10%;
}
main{
    border: 3px solid #B0C4DE;
    height: 80%;
    font-size: 0.8em;
    position: relative;
}

footer{
    border: 3px solid #B0C4DE;
    height:5%;
    font-size: 0.7em;
    background-color: #B0C4DE;
}
body{
    position: relative;
}
button{
    font-size:1em;
    background-color: #B0C4DE;
    border:1px solid #B0C4DE;
}
#bookface{
    position: absolute;
    width: 85%;
    height: 90%;
```

```

border:1px solid #B0C4DE;
background-color: blanchedalmond;
left:6% ;
top: 6% ;
    background-image: url("../images/background.jpg");
background-size: cover;
}
#aid{
    position: absolute;
    border: 3px solid #B0C4DE;
    top:0px;
    left:600px;
}
#typeText{
border: 2px solid #B0C4DE;
padding:0.4em ;
color: gray ;
}
#keyStatus{
    position: absolute;
    border: 2px solid #B0C4DE;
width: 90% ;
    right: 0;
    bottom: 0;
font-size: 0.8em;
padding:0.5em ;
color: gray;

```

(3) JS 部分代码编写如下

```

var myDiv = document.getElementById('bookface');
var images = [
    "../images/1.jpg",
    "../images/2.jpg",
    "../images/3.jpg",
    "../images/4.jpg",
    "../images/5.jpg",
    "../images/6.jpg",
    "../images/7.jpg",
    "../images/8.jpg",
    "../images/9.jpg",
    "../images/10.jpg",
    "../images/11.jpg",
    "../images/12.jpg",
];
//当前位置

```

```

var currentIndex = 0;

document.getElementById("next").onclick = function () {
    currentIndex++;
    if (currentIndex == images.length) currentIndex = 0;
    myDiv.style.backgroundImage = 'url(' + images[currentIndex] +
    ')';
};
document.getElementById("prev").onclick = function () {
    currentIndex--;
    if (currentIndex == -1) currentIndex = images.length - 1;
    myDiv.style.backgroundImage = 'url(' + images[currentIndex] +
    ')';
};

```

### 8.3 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 Edge 浏览器打开项目的结果，如下图 8-3 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 8-4 的二维码，运行测试本项目的第六次开发的阶段性效果。网址为：<https://zyt20020915.github.io/1.6.html>



图 8-3 PC 端运行效果图

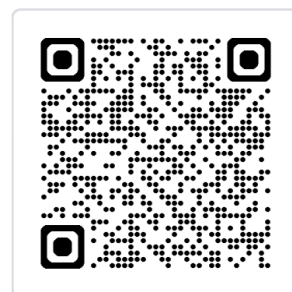


图 8-4 移动端二维码

### 8.4 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，作为项目的第六次迭代，在代码提交和版本管理环节，进入 gitBash 命令行后，按次序输入以下命令：

```

$ cd /d
$ cd webUI
$ git init

```

编写好 phase6.html、mycss6.css 和 myjs6.js 的代码，测试运行成功后，执行下面命令提交代码：

```

$ git add phase/phase6.html mycss6.css mgjs6.js

```

```
$ git commit -m 项目第六版: UI 的个性化键盘交互控制的设计开发
```

成功提交代码后, gitbash 的反馈如下所示:

```
32614@LAPTOP-JSNMSI8H MINGW64 /d/webUI (master)
$ git commit -m 项目第六版: UI的个性化键盘交互控制的设计开发
[master b73b0a6] 项目第六版: UI的个性化键盘交互控制的设计开发
3 files changed, 254 insertions(+)
```

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所示:

```
32614@LAPTOP-JSNMSI8H MINGW64 /d/webUI (master)
$ git log
commit b73b0a61fc47fae44d725c333e79de6cfeebc93e (HEAD -> master)
Author: 江科师大钟雅婷 <3261458337@qq.com>
Date: Thu Jun 6 20:16:08 2024 +0800
```

项目第六版: UI的个性化键盘交互控制的设计开发

## 9. 用 gitBash 工具管理项目的代码仓库和 http 服务器

### 9.1 经典 Bash 工具介绍

当我们谈到命令行时, 我们实际上指的是 shell。shell 是一个程序, 并将它们传递给操作系统执行。几乎所有的 Linux 发行版都提供了一个来自 GNU 项目的 shell 程序。这个名字是伯恩-再次外壳的首字母缩写, 指的是 bash 是 sh 的增强替代品, 最初的 Unix 外壳程序由史蒂夫·伯恩写。和 Windows 一样, 类似 inix 的 Linux 操作系统以分层目录结构组织文件。这意味着它们被组织成一个树状的目录模式(在其他系统中有时称为文件夹), 其中可能包含文件和其他目录。文件系统中的第一个目录称为根目录。根目录包含文件和子目录, 其中包含更多的文件和子目录等等<sup>[6]</sup>。

### 9.2 通过 GitHub 平台实现本项目的全球域名

GitHub 是一个面向开源及私有软件项目的托管平台, 因为只支持 Git 作为唯一的版本库格式进行托管, 故名 GitHub。我们要通过 GitHub 平台实现自己项目的全球域名, 首先在官网注册自己的 GitHub 账号, 然后登录, 如下图 9-1 所示。进入 GitHub 平台后在设置中自定义自己的域名, 比如我的域名是 zyt20020915.github.io。



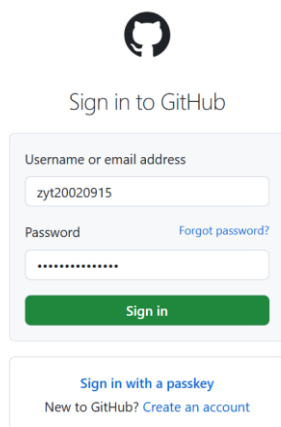


图 9-1 登录 GitHub 账号

### 9.3 创建一个空的远程代码仓库

在自己的 GitHub 平台里，点击下图 9-2 所示的界面图中的绿色按钮，即可进入创建远程代码仓库的界面。点击下图 9-3 的右下角的绿色“Create repository”，则可创建一个空的远程代码仓库。

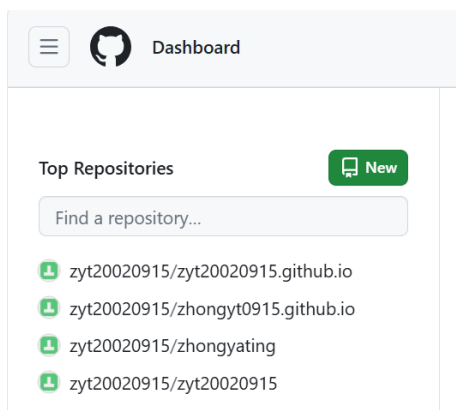


图 9-2 界面图

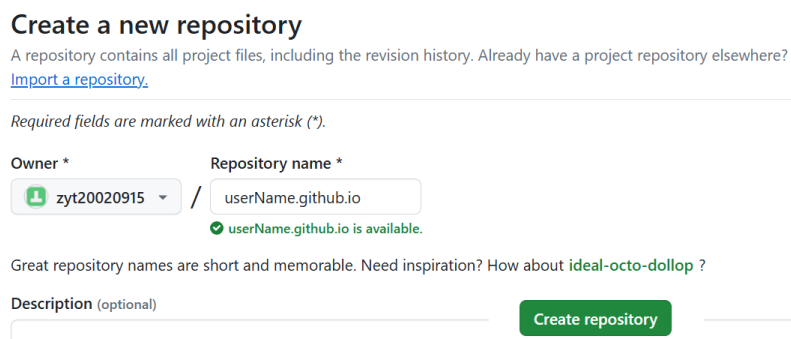


图 9-3 创建远程代码仓库界面图

### 9.4 设置本地仓库和远程代码仓库的链接

进入本地 webUI 项目的文件夹后，通过下面的命令把本地代码仓库与远程建

立密钥链接<sup>[7]</sup>。

```
$ echo "WebUI 应用的远程 http 服务器设置" >> README.md
$ git init
$ git add README.md
$ git commit -m "这是我第一次把代码仓库上传至 gitHub 平台"
$ git branch -M main
$ git remote add origin
https://github.com/zyt20020915/userName.github.io.git
$ git push -u origin main
```

本项目使用 window 平台，gitbash 通过默认浏览器实现密钥生成和记录，第一次链接会要求开发者授权，如下图 9-4 授权界面图所示：

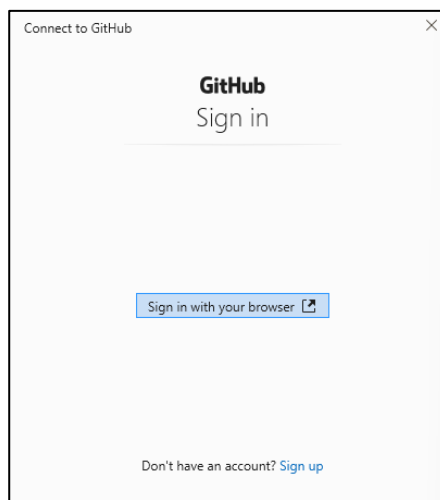


图 9-4 授权界面图

再次确认授权 gitBash 拥有访问改动远程代码的权限，如下图 9-5 所示：

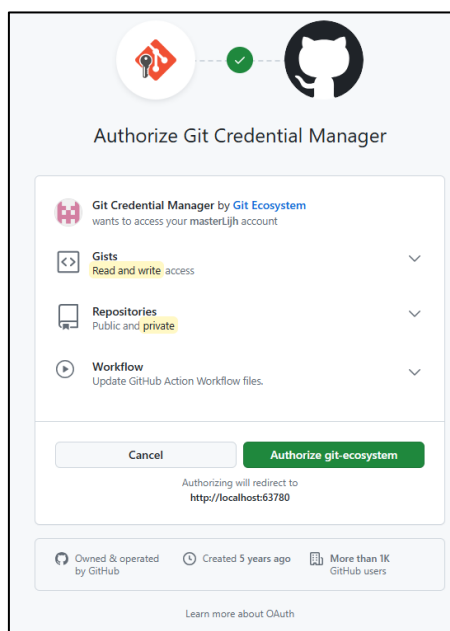


图 9-5 再次确认授权图

最后，GitHub 平台反馈：gitBash 和 gitHub 平台成功实现远程链接如图 9-6 所示。

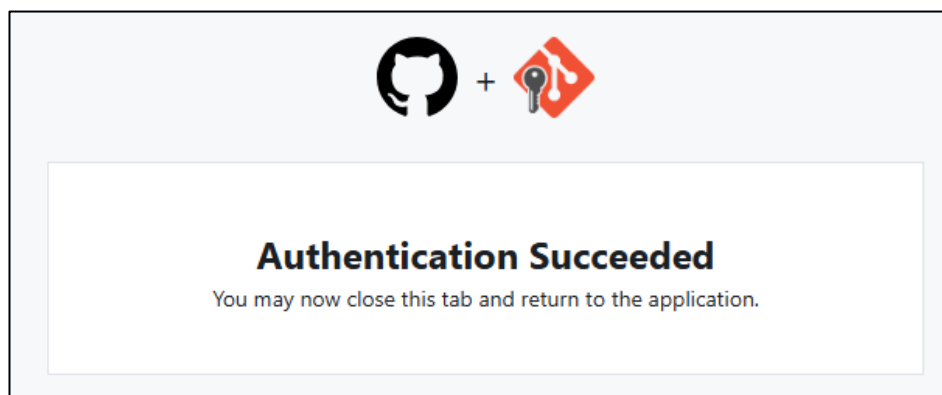


图 9-6 GitHub 平台反馈图

从此，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 github 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，本文截取项目最后呈现的效果图，如图 9-7 所示：

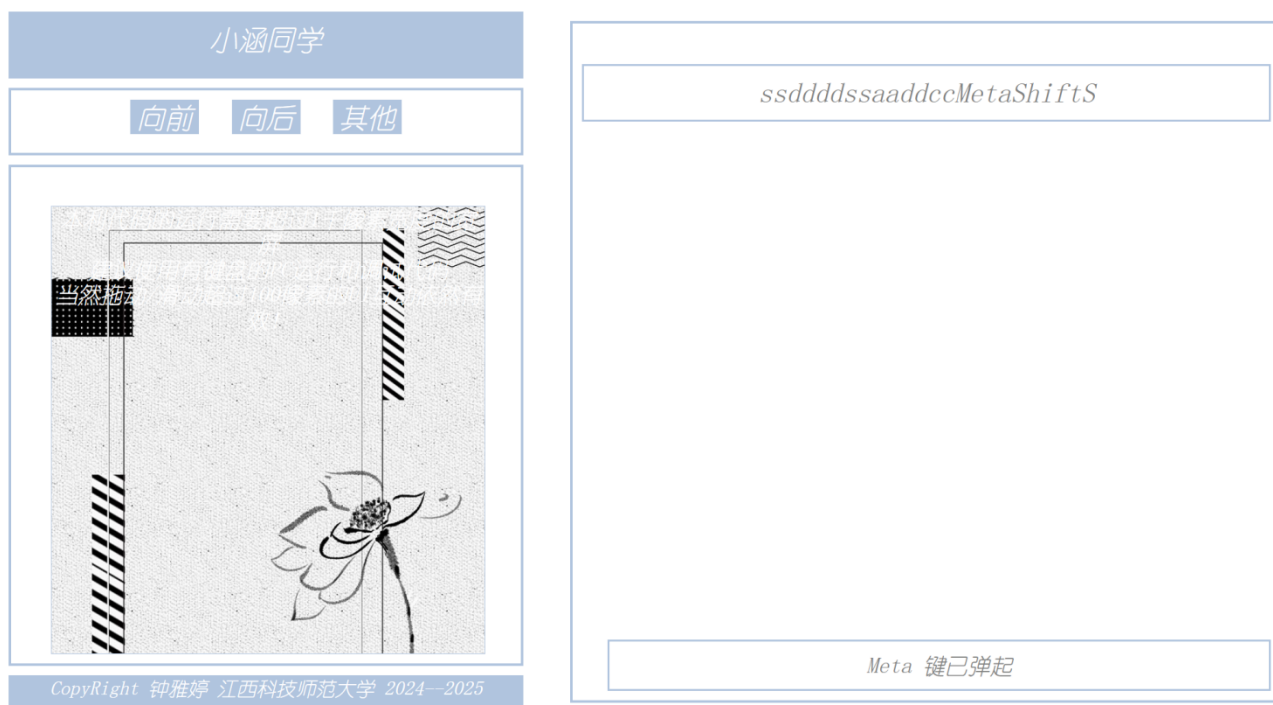


图 9-7 项目效果图

## 参考文献

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