

成绩:

# 江西科技师范大学

## 课程设计（论文）

题目（中文）：基于 Web 客户端技术的个性化 UI 的设计和编程

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

院（系）：元宇宙产业学院

专    业：计算机科学与技术

学生姓名：柴葆明

学    号：20213590

指导教师：李健宏

2024 年 6 月 19 日

# 目录

基于 Web 客户端技术的个性化 UI 的设计和编程 .....	1
( Customized UI design and Programming based on Web client technology ) .....	1
1. 前言 .....	1
1.1 毕设任务分析 .....	1
1.2 研学计划 .....	2
1.3 研究方法 .....	2
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模式 .....	6
3. 内容设计概要 .....	7
3.1 分析和设计 .....	7
3.2 项目的实现和编程 .....	8
3.3 项目的运行和测试 .....	9
3.4 项目的代码提交和版本管理 .....	10
4. 移动互联时代的 UI 开发初步——窄屏终端的响应式设计 .....	11
4.1 分析和设计 .....	11
4.2 项目的实现和编程 .....	12
4.3 项目的运行和测试 .....	13

4.4 项目的代码提交和版本管理 .....	14
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 .....	15
5.3 项目的运行和测试 .....	17
5.4 项目的代码提交和版本管理 .....	17
6. 个性化 UI 设计中对鼠标交互的设计开发 .....	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 .....	19
6.4 项目的代码提交和版本管理 .....	20
7. 对触屏和鼠标的通用交互操作的设计开发 .....	21
7.1 分析和设计 .....	21
7.2 项目的实现和编程 .....	21
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 .....	24
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 .....	26
8.4 项目的代码提交和版本管理 .....	27
9. 谈谈本项目中的高质量代码 .....	27

10. 用 gitBash 工具管理项目的代码仓库和 http 服务器 .....	28
10.1 经典 Bash 工具介绍 .....	28
10.2 通过 gitHub 平台实现本项目的全球域名 .....	28
10.3 创建一个空的远程代码仓库 .....	30
10.4 设置本地仓库和远程代码仓库的链接 .....	30
参考文献: .....	34
写作指导: .....	错误! 未定义书签。

# 基于 Web 客户端技术的个性化 UI 的设计和编程

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

科师大元宇宙产业学院 2021 级 柴葆明

**摘要：**近十年来，基于 HTML5 的 Web 标准在软件开发技术中得到广泛应用，其跨平台和开源特性使其成为各领域应用软件开发的首选。本毕设选取 HTML5 的 Web 客户端技术为主线，深入研究和实践程序设计与软件开发。通过广泛查阅技术书籍、开发者论坛和文献，成功设计开发了一个个性化用户界面（UI）的应用程序。项目中使用 HTML 进行内容建模，CSS 实现 UI 外观设计，JavaScript 编程实现 UI 交互功能，并手动编写每行代码，未使用外部框架或库。响应式设计确保应用智能适配不同移动设备屏幕，面向对象的程序设计思想则构建了通用的指针模型，统一鼠标和触屏控制，实现高质量的代码。工程管理上采用增量式开发模式，通过六次增量重构（A:分析，D:设计，I:实现，T:测试），逐步优化项目。代码管理利用 Git 进行版本控制，经多次提交和测试修改后，最终通过 GitHub 实现全球互联网部署，用户可通过地址或二维码跨平台访问应用。

**关键词：**Web 技术；ADIT 开发模式；UI；GitHub

## 1. 前言

根据个人对毕业设计项目和毕业论文的理解，在下文分别阐述：毕设任务分析、研学计划和研究方法。

### 1.1 毕设任务分析

毕设任务要求学生综合运用本科阶段学习的计算机科学技术知识，尤其是程序设计和软件工程领域学习的方法、训练的代码能力，架构自己感兴趣的技术路线，结合自己探求的问题形成软件需求，然后有条理第系统落实分析问题、建立模型、软件设计、系统实施、测试调试的等传统软件工程的全部的流程。践行毕业设计，总结开发文档撰写论文，践行二者的有机结合。毕业设计的内涵就是大学理论的学习在实践层面做一次综合演练和总结，期间也需要要配合学习当前最新的一些流行技术，在以形成自己对计算机软硬件体系的系统而专业的理解后，再总结撰写毕业论文。同时围绕本专业的目标，选择好自己的方向和技术路线，在导师的指导下，成长为本专业的“四有人才”：1、具有现代工程思维的工程师；

2、掌握现代数字工具的人；3、具有国际视野能自主学习的学习者；4、兼具人文情怀全面发展的人。毕设分为二个阶段完成，首先选择一条自己感兴趣的技术实践路线，把核心的技术加以整合学习，以导师的案例项目为参考，主要是理解好各个技术之间的关系，在项目中的作用和分工，更重要的是在项目实施中提升自己的写高质量的代码能力。当仿造导师的案例的技术基本实现后，则可以视为实践和理论基本打通，此时就可进入第二个阶段，开始真正做自己的毕设软件。第二阶段一般按软件工程的标准来规范开发：1、结合自己的问题做出定义和分析；2、设计一套合适的技术解决方案；3、按解决方案设计流程和编写相关代码，实现技术部署；4、调试代码、测试软件、性能调优。其中第3、4步可以发现前面步骤的问题，因此可能会在第2，3，4步多次循环，发现和解决第2步的设计失误或第3步的代码错误。当然大部分工作是用在第3步的构建代码体系和落实软件架构的具体实施和细节。本科毕设的方案设计的细化优化和写代码具体部署两个步骤交替进行的。前者是工程师落实微观和细节层面，而后者则是设计师的工作，确保宏观层面的设计不偏离需求。在开发期间可以产生大量开发文档，对这些文档做一个总结，再结合本专业的理论就可形成自己的论文，用实践来加深理论理解，进而加深本科期间学习的理论的真实体会。

## 1.2 研学计划

书名	时间
The foundation of computer science	第一、二周
Web techonology	第三、四周
The art of javascript	第五、六周

## 1.3 研究方法

文献法和模型研究法是两种常用的研究方法，它们在科学研究、社会科学研究等多个领域都有广泛的应用。下面将分别对这两种方法进行详细阐述。

### 1.3.1 文献法

文献法，也称为文献综述法或文献研究法，是一种通过收集、分析、评价和综合已有文献资料来获取信息、知识或研究结果的方法。这种方法不直接参与实际的实验或观察，而是通过对现有文献的分析来获取研究所需的材料。

文献法的特点：

(1)历史性：文献法通常涉及对历史文献的回顾，研究者需要了解该领域的历史发展脉络。

(2)间接性：研究者并不直接参与研究对象的活动，而是通过阅读和分析他人的研究成果来获取知识。

(3)综合性：文献法要求研究者能够综合多个来源的信息，并对这些信息进行批判性分析。

文献法的步骤：

(1)确定研究问题：明确研究的目的和问题。

(2)文献搜索：通过图书馆、互联网等渠道搜索相关文献。

(3)文献筛选：根据研究问题和研究需要，筛选出相关的、高质量的文献。

(4)文献分析：对选定的文献进行阅读、分析和评价，提取关键信息。

(5)文献整合：将分析得出的信息整合起来，形成对研究问题的全面理解。

(6)撰写报告：基于文献分析的结果，撰写文献综述报告。

### 1.3.2 模型研究法

模型研究法是指通过建立和分析模型来研究现实世界中的现象或系统的方法。这些模型可以是物理模型、概念模型或数学模型等，其目的是为了简化问题、预测结果或理解特定现象的机制。

模型研究法的特点：

(1)抽象性：模型是对现实世界的抽象和简化，它忽略了一些细节，以便于分析和理解。

(2)预测性：模型可以帮助研究者预测系统的行为，对于指导实践具有重要意义。

(3)验证性：通过与实际观测结果的比较，可以验证模型的有效性和局限性。

模型研究法的步骤：

- (1)确定研究问题：明确研究的目的和需要解决的实际问题。
- (2)建立模型：根据研究问题的特点，选择合适的模型类型，并构建模型。
- (3)参数设定：为模型设定必要的参数和初始条件，以使其能够适用于具体研究对象。
- (4)模型求解：利用数值方法、仿真技术或理论推导等手段，对模型进行求解并得出结果。
- (5)模型评估：对模型的结果进行评估和分析，检查模型的有效性、可靠性和适用性。
- (6)模型优化：根据评估结果对模型进行调整和优化，以提高模型的预测能力和适用性。
- (7)模型解释：解释模型的结果，理解其中蕴含的现象或机制。
- (8)报告撰写：将模型研究的过程、结果和结论进行系统整理，并撰写研究报告或论文。

### 1.3.3 文献法与模型研究法的比较

- 1.数据来源：文献法主要依赖于已有文献资料，而模型研究法则通过建立模型来分析现象。
- 2.研究对象：文献法更适用于对历史回顾、概念分析等研究对象，而模型研究法适用于对系统、现象进行预测、分析和优化。
- 3.独立性：文献法相对独立，主要依赖于文献资料，而模型研究法需要研究者建立自己的模型并进行求解。
- 4.深度与广度：文献法可以提供对研究问题的广泛理解和综合性分析，而模型研究法则能够深入探讨模型背后的物理机制和参数影响。

## 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。

### 2.1.1 发展历史

1989 年，蒂姆·伯纳斯-李爵士发明了万维网（见原始提案）。1990 年 10 月，他创造了“万维网”一词，编写了第一台万维网服务器“httpd”和第一个客户端程序（浏览器和编辑器）“万维网”。

他编写了“超文本标记语言”（HTML）的第一个版本，这是一种具有超文本链接功能的文档格式化语言，成为网络的主要发布格式。随着 Web 技术的普及，他对 URI、HTTP 和 HTML 的最初规范得到了改进，并在更大的圈子里进行了讨论。

### 2.1.2 万维网联盟

1994 年，在许多公司的敦促下，成立了万维网联盟，向网络投入了越来越多的资源。Tim Berners-Lee 爵士开始领导网络联盟团队的重要工作，以促进一致的架构，适应网络标准的快速发展，从而构建网站、浏览器和设备，体验网络所提供的一切。

蒂姆·伯纳斯-李爵士在创建万维网联盟时创建了一个同行社区。Web 技术已经发展得如此之快，因此组建一个单一的组织来协调 Web 标准至关重要。蒂姆接受了麻省理工学院的邀请，他在联盟方面有丰富的经验，主持 W3C。他从一开始就要求 W3C 具有全球影响力。

### 2.1.3 Web 平台与 Web 编程

让我们先简单介绍一下 Web，它是万维网的缩写。大多数人说“网络”而不是“万维网”，我们将遵循这一惯例。网络是一组文档，称为网页，由世界各地的计

计算机用户共享（大部分）。不同类型的网页做不同的事情，但至少，它们都在电脑屏幕上显示内容。所谓“内容”，我们指的是文本、图片和用户输入机制，如文本框和按钮。<sup>[2]</sup>

Web 编程是一个很大的领域，通过不同的工具实现不同类型的 Web 编程。所有的工具都使用核心语言 HTML，所以几乎所有的网络编程书籍都在某种程度上描述了 HTML。这本教科书涵盖了 HTML5、CSS 和 JavaScript，所有这些都是深入的。众所周知，这三种技术是客户端 web 编程的支柱。使用客户端 web 编程，所有网页计算都在最终用户的计算机（客户端计算机）上执行。<sup>[3]</sup> 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）<sup>[3]</sup>；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。

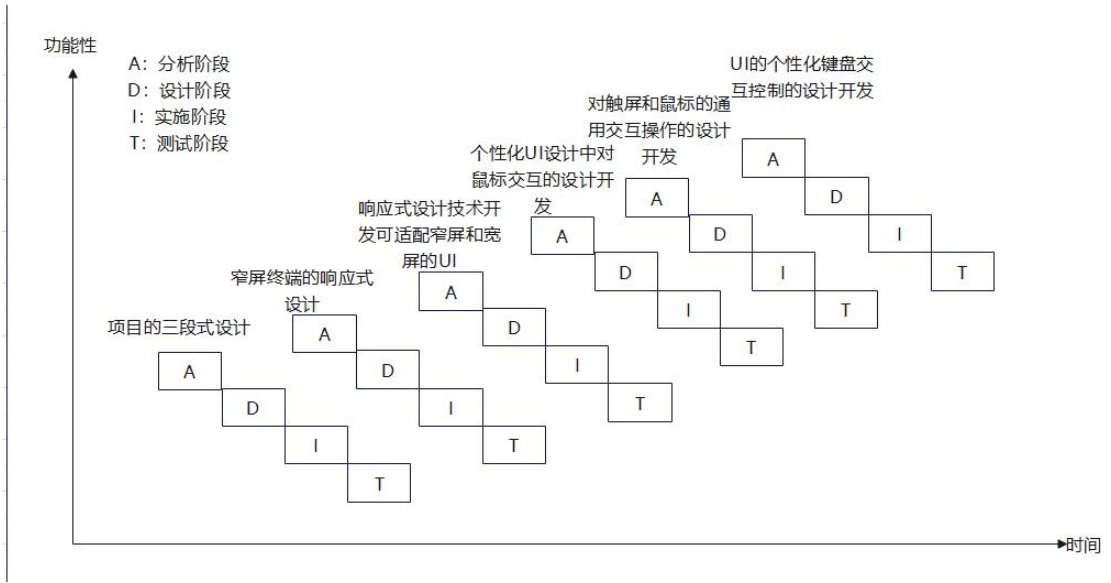
## 2.2 项目的增量式迭代开发模式

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 1 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。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<sup>[5]</sup>。本项目中我一共做了六次项目的开发迭代，如下图 2-1 所示：



### 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 用例图所示：

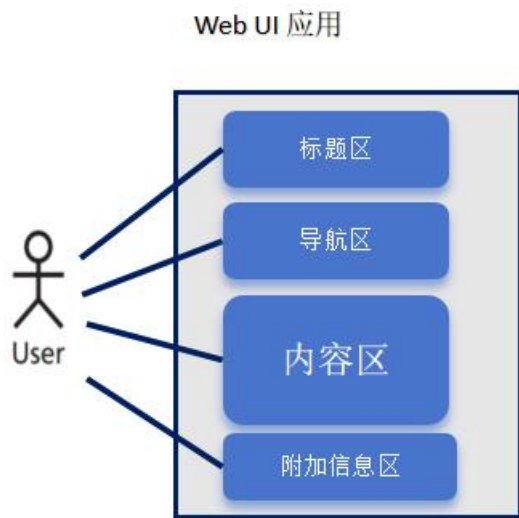


图 3-1 用例图

## 3.2 项目的实现和编程

### 一、HTML 代码编写如下：

```
<header>
    《计算思维和语言学习》

    <nav>
        <button>向前</button>
        <button>向后</button>
    </nav>
</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>
    <p>
    Copyright from 柴葆明 江西科技师范大学 2022--2025
    </p>
</footer>
```

### 二、CSS 代码编写如下：

```
*{
```

```

margin: 10px;
text-align: center;
}
header{
height: 200px;
border: 2px solid rgb(0, 204, 255);
}
nav{
height: 40px;
border: 2px solid rgb(0, 204, 255);
}
main{
height: 200px;
border: 2px solid rgb(0, 204, 255);
}
footer{
height: 100px;
border: 2px solid rgb(0, 204, 255);
}

```

### 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，如下图 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 3-3 的二维码，运行测试本项目的第一次开发的阶段性效果。



图 3-2 PC 端运行效果图



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 江科师大柴葆明  
$ git config user.email 1226400765@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：

```
$ git commit -m 项目第一版：“三段式”式的内容设计概要开发  
[master (root-commit) fcab9df] 项目第一版：“三段式”式的内容设计概要开发  
1 file changed, 36 insertions(+)  
create mode 100644 index.html
```

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：

```
commit fcab9df5d0986870aaa6050070e581037a00ad76 (HEAD -> master)
Author: 江科师大柴葆明 <1226400765@qq.com>
Date: Wed Jun 12 14:21:15 2024 +0800
```

项目第一版：“三段式”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

移动互联网时代人们使用智能手机浏览日益频繁，对网页的浏览不止于 PC 平台，因此应设计 PC 平台和手机平台的动态响应来解决用户需求。

用 JavaScript 动态读取显示设备的信息，然后按设计，使用 js+css 来部署适配当前设备的显示的代码。为软件的四个区域分配了高度的比率，如图 4-1 用例图所示。通过把设备的高度全部分配给 body 对象，结合前面的高度分配实现各区域响应式设计；为应用计算了基础字体的大小，并利用了 body 的遗传机制，结合 css 对字体的相对控制，实现了字体的响应式设计。

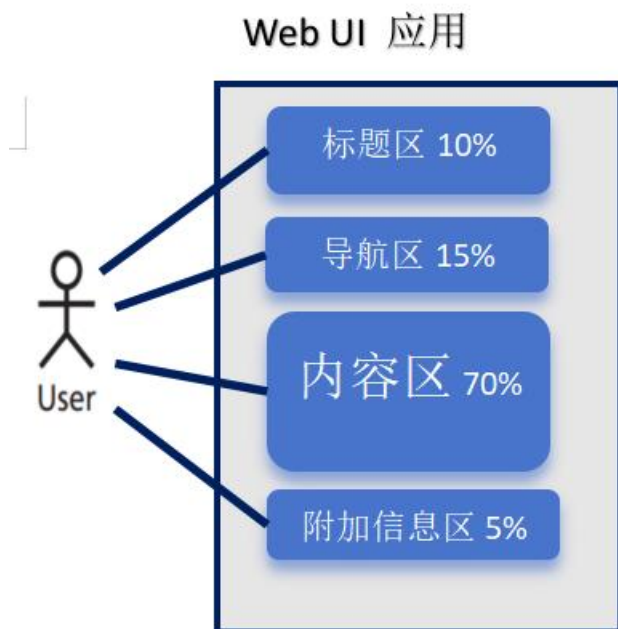


图 4-1 用例图

## 4.2 项目的实现和编程

### 实现代码

用汉语言来描述我们是如何实现的，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。如代码块 4-1 所示：

```
*{
    margin: 10px;
    text-align: center;
}
header{
    height: 15%;
    border: 2px solid rgb(0, 204, 255);
}
nav{
    height: 10%;
    border: 2px solid rgb(0, 204, 255);
}
main{
    height: 70%;
    border: 2px solid rgb(0, 204, 255);
}
footer{
    height: 5%;
    border: 2px solid rgb(0, 204, 255);
}
```

代码块 4-1

用汉语言来描述我们是如何实现的：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。如代码块 4-2 所示：

```
<script>
    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";
</script>

```

代码块 4-2

### 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，本文此处仅给出 PC 端用 Chrome 浏览器打开项目的结果，如下图 4-1 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 4-2 的二维码，运行测试本项目的第二次开发的阶段性效果。

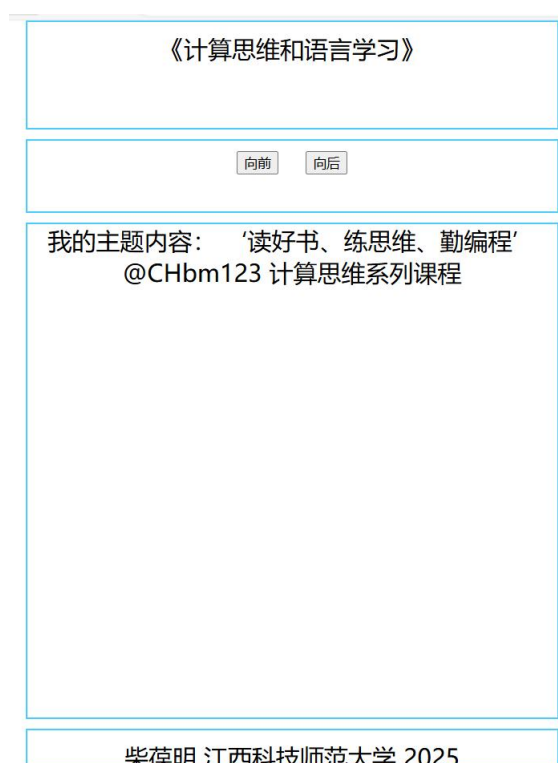


图 4-1 PC 端运行效果图



图 4-2 移动端二维码

## 4.4 项目的代码提交和版本管理

成功提交代码后，gitbash 的反馈如下所示：

```
$ git commit -m 项目第二版:移动互联网时代人们使用智能手机浏览日益频繁，对网页的浏览不止于PC平台，因此应设计PC平台和手机平台的动态响应来解决用户需求。
[master 55a6695] 项目第二版:移动互联网时代人们使用智能手机浏览日益频繁，对网页的浏览不止于PC平台，因此应设计PC平台和手机平台的动态响应来解决用户需求。
1 file changed, 18 insertions(+), 2 deletions(-)
```

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：

```
commit 55a6695a9c8b8f78534a7fde296152e77582aca5 (HEAD -> master)
Author: 江科师大柴葆明 <1226400765@qq.com>
Date: Wed Jun 12 14:50:04 2024 +0800
```

```
项目第二版:移动互联网时代人们使用智能手机浏览日益频繁，对网页的浏览不止于PC平台，因此应设计PC平台和手机平台的动态响应来解决用户需求。
```

## 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

进一步完善了对宽屏和窄屏的识别，区分出是 PC 平台或是手机平台，PC 端增加了对鼠标拖动的响应和键盘的响应区，如图 5-1 用例图所示。

对整体宽度进行判断，窄于一定的宽度则判定为手机平台不显示用户键盘响应区。同时利用 JavaScript 的动态读取，为书的封面区域和用户键盘响应区添加监听，监听 mousemove 和 keypress 以此来监督用户的鼠标位置和键盘输入。手机平台则利用 mousedown 暂时来定位用户手指位置。

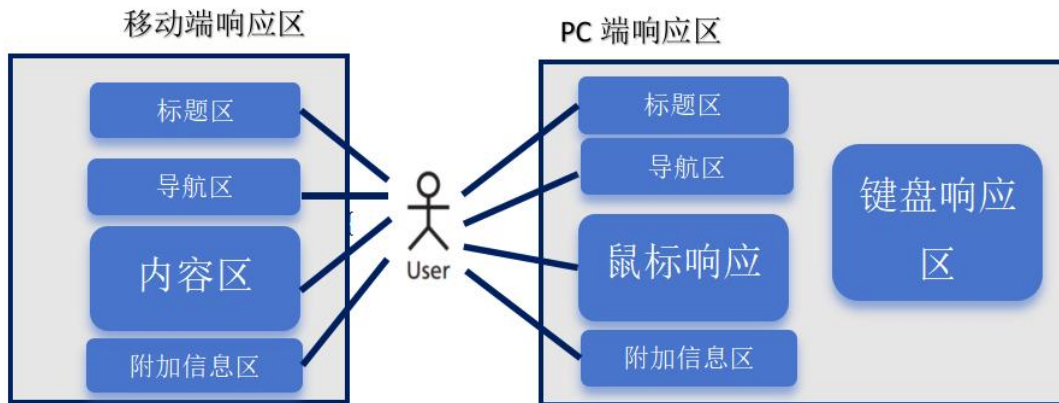


图 5-1 用例图

## 5.2 项目的实现和编程

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

let baseFont = UI.appWidth /20;
//通过改变 body 对象的字体大小，这个属性可以影响其后代
document.body.style.fontSize = baseFont + "px";
//通过把 body 的高度设置为设备屏幕的高度，从而实现纵向全屏
//通过 CSS 对子对象百分比（纵向）的配合，从而达到我们响应式设计的目标
document.body.style.width = UI.appWidth + "px";
document.body.style.height = UI.appHeight - 62 + "px";
if(window.innerWidth<1000){
    $("aid").style.display='none';
}
$("aid").style.width=window.innerWidth-UI.appWidth-30+'px';
$("aid").style.height= UI.appHeight-62+'px';

//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+")";
});
$("#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 $

</script>

```

### 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 Chrome 浏览器打开项目的结果，如下图 5-2 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 5-3 的二维码，运行测试本项目的第三次开发的阶段性效果。

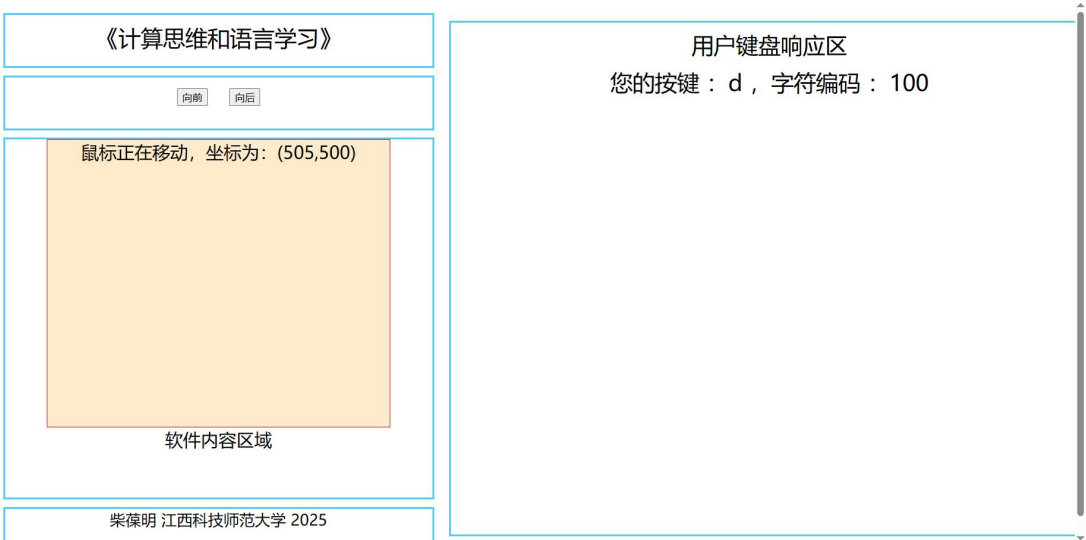


图 5-2 PC 端运行效果图



图 5-3 移动端二维码

### 5.4 项目的代码提交和版本管理

成功提交代码后，gitbash 的反馈如下所示：

```
沐伶@LAPTOP-NUMPC7T2 MINGW64 /webUI (master)
$ git commit -m 应用响应式设计技术开发可适配窄屏和宽屏的UI,进一步完善了对宽屏和窄屏的识别,区分出是PC平台或是手机平台, PC端增加了对鼠标拖动的响应和键盘的响应区
[master 68d43ed] 应用响应式设计技术开发可适配窄屏和宽屏的UI,进一步完善了对宽屏和窄屏的识别,区分出是PC平台或是手机平台, PC端增加了对鼠标拖动的响应和键盘的响应区
1 file changed, 85 insertions(+), 21 deletions(-)
```

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：

```
commit 68d43edb63910423868933ec57da070210e54530 (HEAD -> master)
Author: 江科师大柴葆明 <1226400765@qq.com>
Date: Thu Jun 13 10:42:58 2024 +0800
```

应用响应式设计技术开发可适配窄屏和宽屏的UI,进一步完善了对宽屏和窄屏的识别,区分出是PC平台或是手机平台, PC端增加了对鼠标拖动的响应和键盘的响应区

## 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

对鼠标模型进一步分析添加 mousedown 事件监听鼠标按下的时机,记录下元素的初始位置,并开始监听 mousemove 事件,随着鼠标的移动,相应地改变元素的位置。当用户释放鼠标按钮时, mouseup 事件被触发,如果移动到其他元素则触发 mouseout 事件,此时可以根据鼠标释放的位置来确定最终的拖动位置,更新元素的位置、完成拖拽操作。用例图如图 6-1 所示。

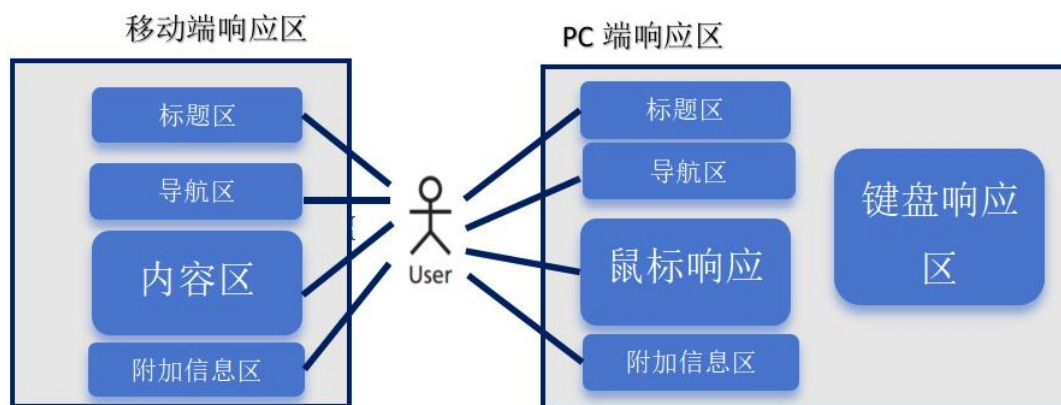


图 6-1 用例图

### 6.2 项目的实现和编程

```
var mouse={};
mouse.isDown= false;
mouse.x= 0;
mouse.y= 0;
mouse.deltaX=0;
$("#bookface").addEventListener("mousedown",function(ev){
    mouse.isDown=true;
    mouse.x= ev.pageX;
    mouse.y= ev.pageY;
    console.log("mouseDown at x: "+(""+mouse.x +"," +mouse.y +"))");
```

```

$("bookface").textContent= "鼠标按下，坐标： "+(""+mouse.x+","+mouse.y+"");
});
$("bookface").addEventListener("mouseup",function(ev){
    mouse.isDown=false;

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

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

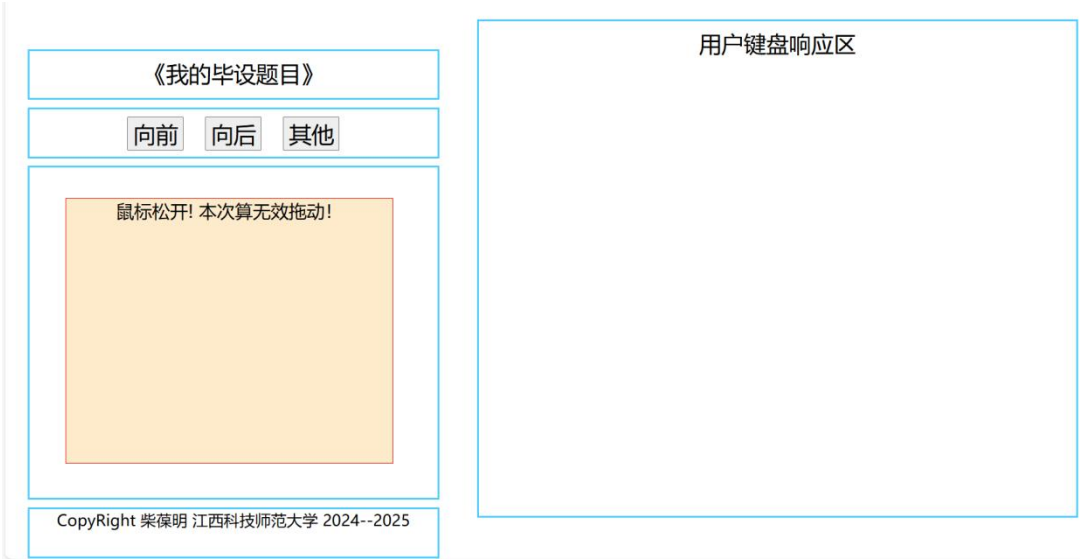
```

## 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 Chrome 浏览器打开项目的结果，如下图 6-1 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 6-2 的二维码，运行测试本项目的第四次开发的阶段性效果。



6-1 PC 端运行图



6-2 移动端二维码

## 6.4 项目的代码提交和版本管理

成功提交代码后，gitbash 的反馈如下所示：

```
PKT-CLAPTOP-NUMPC712-MINGW64 /web01 (master)
$ git commit -m 项目第四版:对鼠标模型进一步分析添加mousedown事件监听鼠标按下的时机，记录下元素的初始位置，并开始监听mousemove事件，随着鼠标的移动，相应地改变元素的位置。当用户释放鼠标按钮时，mouseup事件被触发，如果移动到其他元素则触发mouseout事件，此时可以根据鼠标释放的位置来确定最终的拖动位置，更新元素的位置、完成拖拽操作。
[master 8f756b6] 项目第四版:对鼠标模型进一步分析添加mousedown事件监听鼠标按下的时机，记录下元素的初始位置，并开始监听mousemove事件，随着鼠标的移动，相应地改变元素的位置。当用户释放鼠标按钮时，mouseup事件被触发，如果移动到其他元素则触发mouseout事件，此时可以根据鼠标释放的位置来确定最终的拖动位置，更新元素的位置、完成拖拽操作。
1 file changed, 193 insertions(+), 116 deletions(-)
rewrite index.html (80%)
```

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：



```
commit 8f756b6cbb7335e37812534070559e3a796b701d (HEAD -> master)
Author: 江科师大柴葆明 <1226400765@qq.com>
Date: Sun Jun 16 20:21:54 2024 +0800

项目第四版:对鼠标模型进一步分析添加mousedown事件监听鼠标按下的时机，记录下元素的初始位置，并开始监听mousemove事件，随着鼠标的移动，相应地改变元素的位置。当用户释放鼠标按钮时，mouseup事件被触发，如果移动到其他元素则触发mouseout事件，此时可以根据鼠标释放的位置来确定最终的拖动位置，更新元素的位置、完成拖拽操作。
```

## 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

本项目追求的是 PC 平台和手机平台双端浏览，所以在进一步完善鼠标建模的基础上应完善触屏建模，判断用户是触屏或是键盘在对鼠标模型的基础上，添加触屏模型，分别对 touchstart、touchend、touchmove 添加监听，在手机上实现同 PC 平台一致的拖动功能，满足用户的需求。如图 7-1 用例图所示

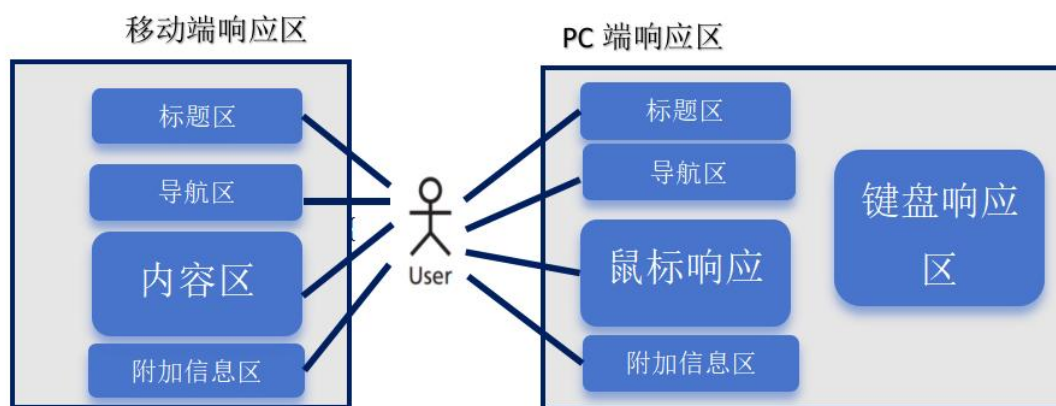


图 7-1 用例图

### 7.2 项目的实现和编程

实现代码,将项目针对于 PC 端和移动端分为触屏事件和鼠标事件进行监视，判断是触屏事件还是鼠标事件，同时将代码分为三部分分别监视事件的开始、结束和移动距离，如代码块 7-1 监视事件的开始。代码块 7-2 监视事件的结束，代码块 7-3 监视事件的移动。

```

{ //Code Block begin
  let handleBegin = function(ev){
    Pointer.isDown=true;

    if(ev.touches){console.log("touches1"+ev.touches);
      Pointer.x = ev.touches[0].pageX ;
      Pointer.y = ev.touches[0].pageY ;
      console.log("Touch begin : "+"("+Pointer.x +"," +Pointer.y +")" );
      $("bookface").textContent= "触屏事件开始, 坐标: "+"("+Pointer.x+","+Pointer.y+")";
    }else{
      Pointer.x= ev.pageX;
      Pointer.y= ev.pageY;
      console.log("PointerDown at x: "+"("+Pointer.x +"," +Pointer.y +")" );
      $("bookface").textContent= "鼠标按下, 坐标: "+"("+Pointer.x+","+Pointer.y+")";
    }
  };
};

```

代码块 7-1

```

let handleEnd = function(ev){
  Pointer.isDown=false;
  ev.preventDefault()
  //console.log(ev.touches)
  if(ev.touches){
    $("bookface").textContent= "触屏事件结束!";
    if(Math.abs(Pointer.deltaX) > 100){
      $("bookface").textContent += "，这是有效触屏滑动！" ;
    }else{
      $("bookface").textContent += " 本次算无效触屏滑动！" ;
      $("bookface").style.left = '7%' ;
    }
  }else{
    $("bookface").textContent= "鼠标松开!";
    if(Math.abs(Pointer.deltaX) > 100){
      $("bookface").textContent += "，这是有效拖动！" ;
    }else{
      $("bookface").textContent += " 本次算无效拖动！" ;
      $("bookface").style.left = '7%' ;
    }
  }
};

```

代码块 7-2

```

let handleMoving = function(ev){
  ev.preventDefault();

```

```

if (ev.touches){
    if (Pointer.isDown){
        console.log("Touch is moving");
        Pointer.deltaX = parseInt( ev.touches[0].pageX - Pointer.x );
        $("bookface").textContent= "正在滑动触屏，滑动距离： " + Pointer.deltaX +"px 。 ";
        $('bookface').style.left = Pointer.deltaX + 'px' ;
    }
} else{
    if (Pointer.isDown){
        console.log("Pointer isDown and moving");
        Pointer.deltaX = parseInt( ev.pageX - Pointer.x );
        $("bookface").textContent= "正在拖动鼠标，距离： " + Pointer.deltaX +"px 。 ";
        $('bookface').style.left = Pointer.deltaX + 'px' ;
    }
}
};

```

代码块 7-3

## 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 Chrome 浏览器打开项目的结果，如下图 7-1 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 7-2 的二维码，运行测试本项目的第五次开发的阶段性效果。

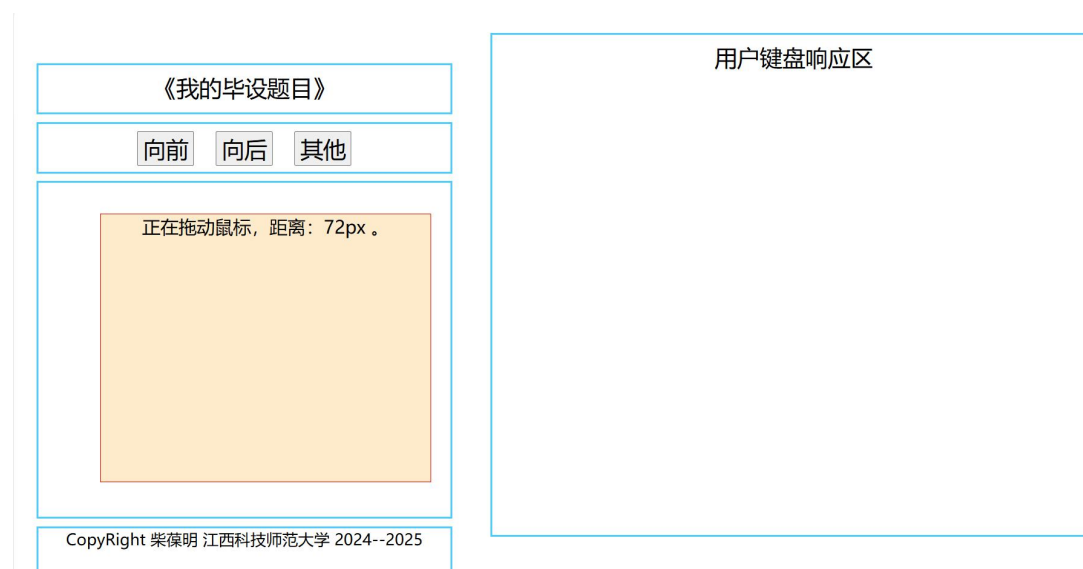


图 7-1 PC 端运行效果图



图 7-2 移动端二维码

## 7.4 项目的代码提交和版本管理

成功提交代码后，gitbash 的反馈如下所示：

```
沐伶@LAPTOP-NUMPC7T2 MINGW64 /webUI (master)
$ git commit -m 项目第五版:进一步完善鼠标建模的基础上应完善触屏建模，判断用户是触屏或是键盘在对鼠标模型的基础上，添加触屏模型，分别对touchstart、touchend、touchmove添加监听，在手机上实现同PC平台一致的拖动功能，满足用户的需求。
[master f6dc94f] 项目第五版:进一步完善鼠标建模的基础上应完善触屏建模，判断用户是触屏或是键盘在对鼠标模型的基础上，添加触屏模型，分别对touchstart、touchend、touchmove添加监听，在手机上实现同PC平台一致的拖动功能，满足用户的需求。
1 file changed, 70 insertions(+), 48 deletions(-)
```

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：

```
commit f6dc94f29159c93cdcc9c293f9990ddc41729217 (HEAD -> master)
Author: 江科师大柴葆明 <1226400765@qq.com>
Date: Sun Jun 16 20:23:46 2024 +0800

    项目第五版:进一步完善鼠标建模的基础上应完善触屏建模，判断用户是触屏或是键盘在对鼠标模型的基础上，添加触屏模型，分别对touchstart、touchend、touchmove添加监听，在手机上实现同PC平台一致的拖动功能，满足用户的需求。
```

## 8. UI 的个性化键盘交互控制的设计开发

### 8.1 分析和设计

利用 keydown 和 keyup 两个底层事件，代替 keypress 实现同时输出按键状态和文本内容并且对特殊字符和符号添加键盘识别功能。完善键盘控制的功能。如图 8-1 用例图所示

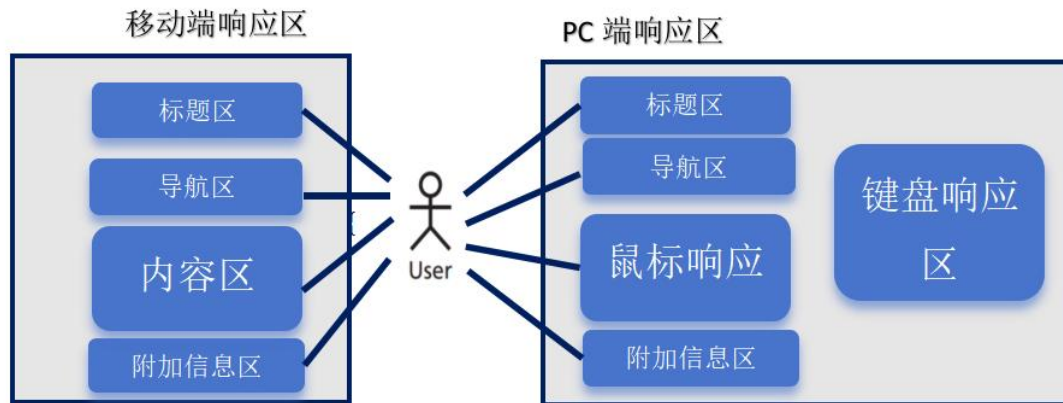


图 8-1 用例图

## 8.2 项目的实现和编程

因为系统中只有一个键盘，所以我们在部署代码时，把键盘事件的监听设置在 DOM 文档最大的可视对象——body 上，通过测试，不宜把键盘事件注册在 body 内部的子对象中。代码如下所示：

```
$("#body").addEventListener("keydown",function(ev){
    ev.preventDefault(); //增加“阻止事件对象的默认事件后”，不仅 keypress 事件将不再响应，而且系统的热键，如“F5 刷新页面/Ctrl+R”、“F12 打开开发者面板”等也不再被响应
    let k = ev.key;
    let c = ev.keyCode;
    $("#keyStatus").textContent = "按下键 : " + k + " , "+"编码 : " + c;
});
```

```
$("#body").addEventListener("keyup",function(ev){
    ev.preventDefault();
    let key = ev.key;
    $("#keyStatus").textContent = key + " 键已弹起";
    if (printLetter(key)){
        $("#typeText").textContent += key;
    }
}
```

```
function printLetter(k){
    if (k.length > 1){ //学生须研究这个逻辑的作用
        return false;
    }
}
```



## 8.4 项目的代码提交和版本管理

成功提交代码后，gitbash 的反馈如下所示：

```
沐伶@LAPTOP-NUMPC7T2 MINGW64 /webUI (master)
$ git commit -m 项目第六版：利用keydown和keyup两个底层事件，代替keypress实现同时
输出按键状态和文本内容并且对特殊字符和符号添加键盘识别功能。完善键盘控制的功能。
[master c44dcb9] 项目第六版：利用keydown和keyup两个底层事件，代替keypress实现同
时输出按键状态和文本内容并且对特殊字符和符号添加键盘识别功能。完善键盘控制的功能
1 file changed, 68 insertions(+), 15 deletions(-)
```

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：

```
commit c44dcb953b2ef7a2511a033bcdffb3228d418e2 (HEAD -> master)
Author: 江科师大柴葆明 <1226400765@qq.com>
Date: Sun Jun 16 20:25:45 2024 +0800

    项目第六版：利用keydown和keyup两个底层事件，代替keypress实现同时输出按键状态
和文本内容并且对特殊字符和符号添加键盘识别功能。完善键盘控制的功能。
```

## 9. 谈谈本项目中的高质量代码

创建一个 Pointer 对象，践行 MVC 设计模式，设计一套代码同时对鼠标和触屏实现控制。

面向对象思想，封装，抽象，局部变量，函数式编程，逻辑。

```
var Pointer = {};
Pointer.isDown= false;
Pointer.x = 0;
Pointer.deltaX=0;
{ //Code Block Begin
    let handleBegin = function(ev){
        Pointer.isDown=true;

        if(ev.touches){console.log("touches1"+ev.touches);
            Pointer.x = ev.touches[0].pageX ;
            Pointer.y = ev.touches[0].pageY ;
            console.log("Touch begin : "+"("+Pointer.x +"," +Pointer.y +")" );
            $("bookface").textContent= "触屏事件开始，坐标： "+"("+Pointer.x+"," +Pointer.y+");";
        }else{
            Pointer.x= ev.pageX;
            Pointer.y= ev.pageY;
```



```
console.log("PointerDown at x: "+"("+Pointer.x +"," +Pointer.y +")" );  
$("bookface").textContent= "鼠标按下，坐标: "+"("+Pointer.x+"," +Pointer.y+")";  
}  
};
```

## 10. 用 gitBash 工具管理项目的代码仓库和 http 服务器

### 10.1 经典 Bash 工具介绍

当我们谈到命令行时，我们实际上指的是 shell。shell 是一个接受键盘命令并将其传递给操作系统执行的程序。几乎所有的 Linux 发行版都提供了一个名为 bash 的 GNU 项目的 shell 程序。这个名字是 bourne-reagain shell 的首字母缩写，指的是 bash 是由 Steve bourne 编写的原始 Unix shell 程序 sh 的增强替代品。<sup>[7]</sup>

像 Windows 一样，像 Linux 这样的类 Unix 操作系统将其文件组织在所谓的分层目录结构中。这意味着它们以树状目录模式组织（有时在其他系统中称为文件夹），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文件夹，依此类推。<sup>[7]</sup>

### 10.2 通过 gitHub 平台实现本项目的全球域名

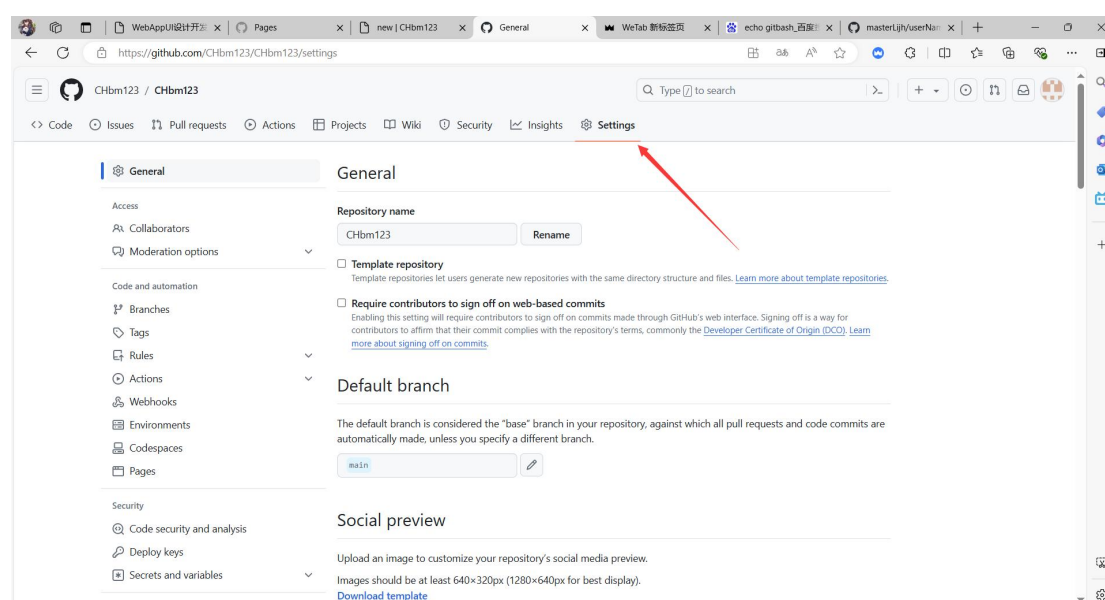


图 10-1 操作 1



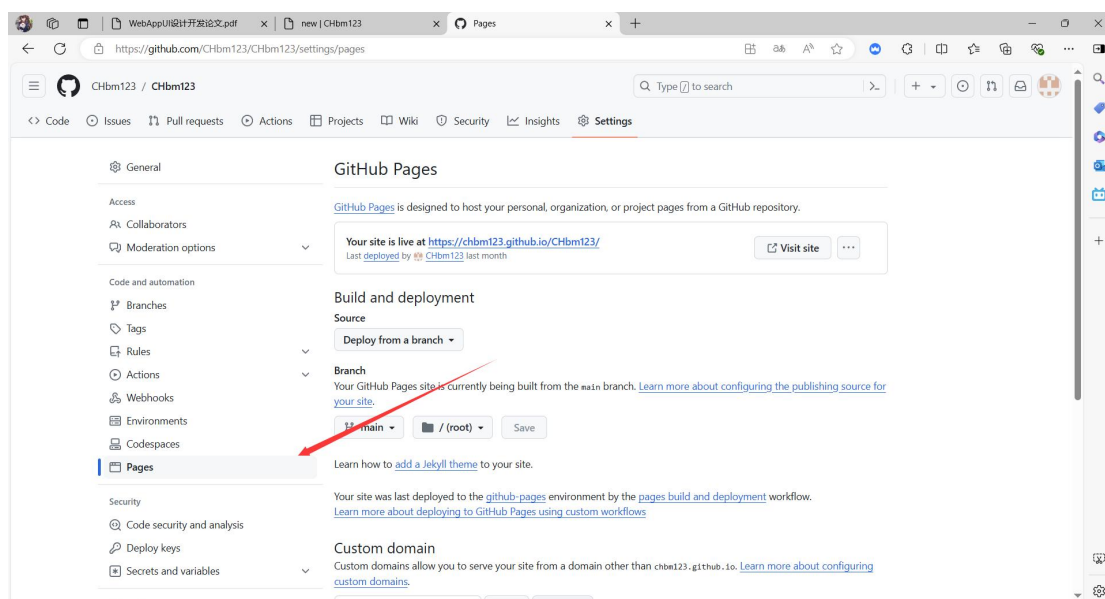


图 10-2 操作 2

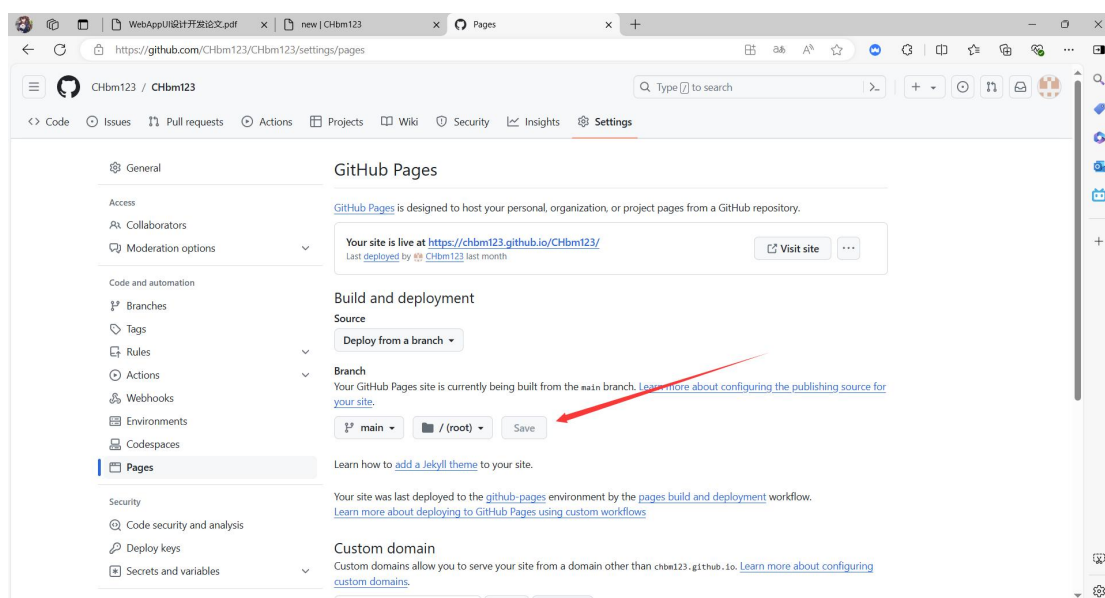


图 10-3 操作 3

## GitHub Pages

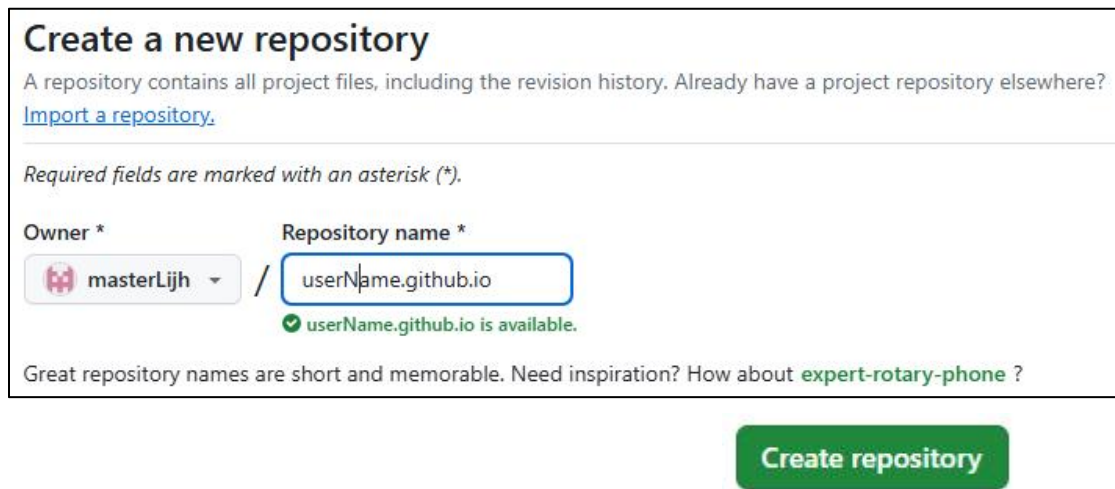
GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://chbm123.github.io/CHbm123/>  
Last deployed by CHbm123 last month

Visit site

图 10-4 操作 4

## 10.3 创建一个空的远程代码仓库




**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \*      Repository name \*

 masterLijh /

✓ userName.github.io is available.

Great repository names are short and memorable. Need inspiration? How about **expert-rotary-phone** ?

**Create repository**

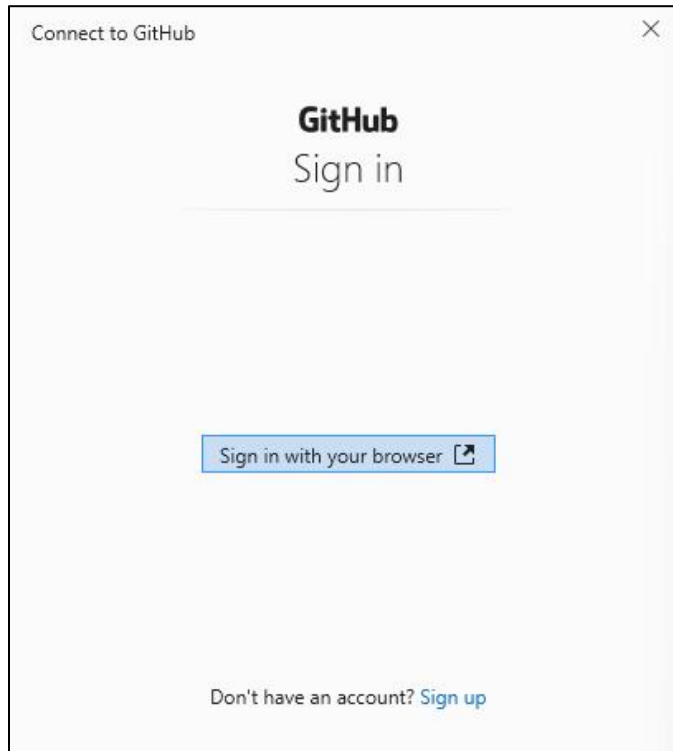
点击窗口右下角的绿色“Create repository”，则可创建一个空的远程代码仓库。

## 10.4 设置本地仓库和远程代码仓库的链接

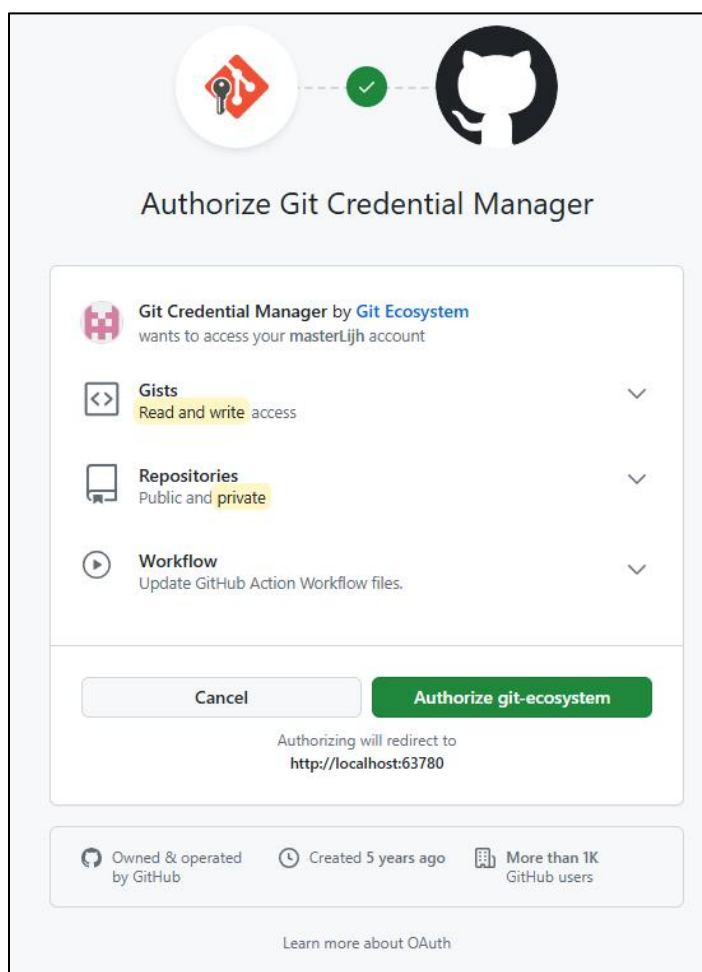
进入本地 webUI 项目的文件夹后，通过下面的命令把本地代码仓库与远程建立密钥链接

```
$ 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/CHbm123/CHbm123
$ git push -u origin main
```

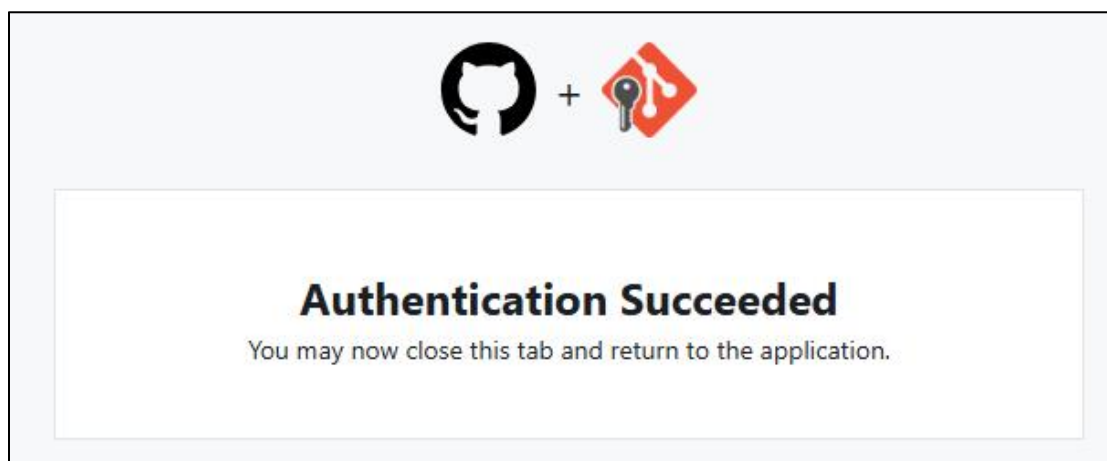
本项目使用 window 平台，gitbash 通过默认浏览器实现密钥生成和记录，第一次链接会要求开发者授权，如下图所示：



再次确认授权 gitBash 拥有访问改动远程代码的权限，如下图所示：



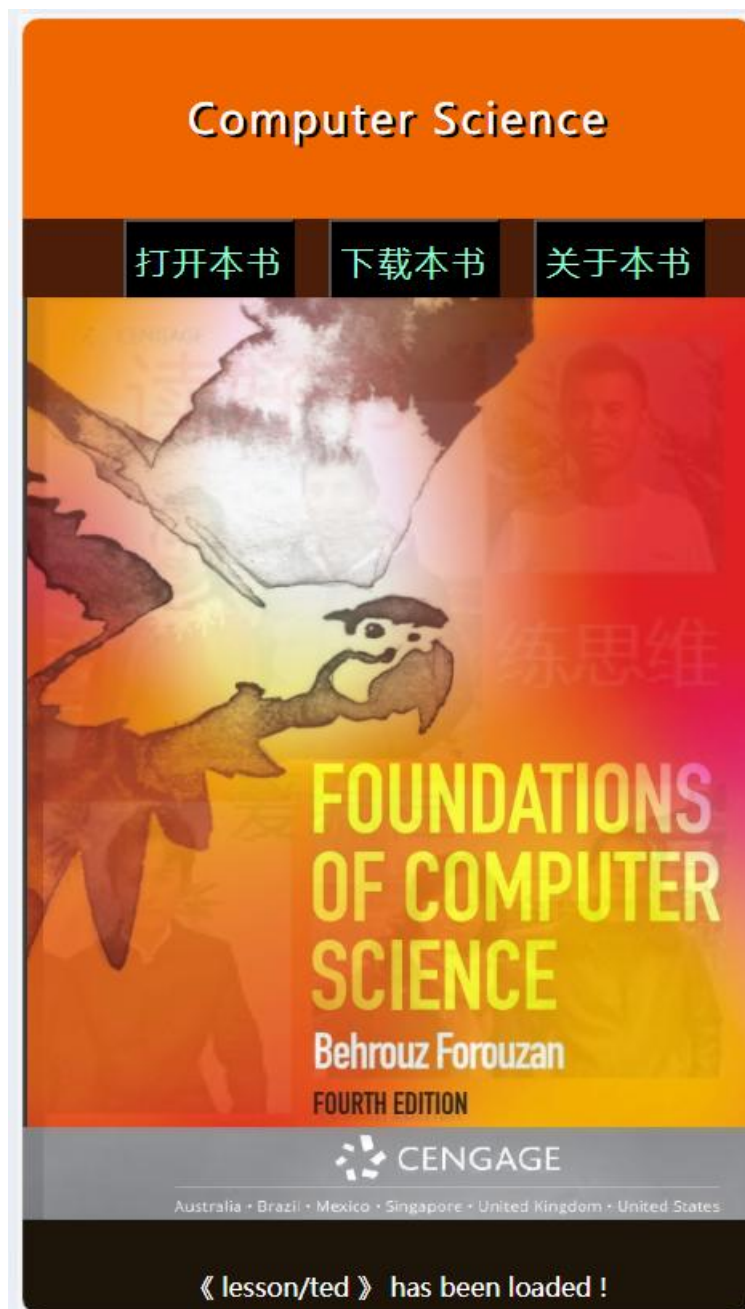
最后，GitHub 平台反馈：gitBash 和 gitHub 平台成功实现远程链接。



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 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，本文截取操作中间

的效果图，如下所示：



全文完成，谢谢！

## 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