

成绩:

江西科技师范大学

毕业设计（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）：Web client based customized UI design and Programming

院（系）：元宇宙产业学院

专 业：计算机科学与技术

学生姓名：刘鑫

学 号：20213578

指导教师：李健宏

2024 年 6 月 18 日

目录

第 1 章 前言	2
1.1 学习计划	2
1.1.1 第一周	2
1.1.2 第二周	2
1.1.3 第三周	2
1.1.4 第四周	2
1.1.5 第五周	2
1.1.6 第六周	2
1.3 研究方法	2
第 2 章 Web 平台和客户端技术概述和文献综述	3
第 3 章 项目的增量式迭代开发模式	4
第 4 章 内容设计概要	5
4.1 分析和设计	5
4.2 项目的实现和编程	6
4.3 项目的运行和测试	7
4.4 项目的代码提交和版本管理	8
第 5 章 移动互联时代的响应式设计和窄屏代码实现	9
第 6 章 适用移动互联时代的响应式设计	11
第 7 章 个性化 UI 设计中鼠标模型	13
第 8 章 通用的 UI 设计	15
第 9 章 UI 的个性化键盘控制	17
第 10 章 本项目中的高质量代码	21
第 11 章 用 gitBash 工具管理本项目的 http 服务器	24
11.1 经典 Bash 工具介绍	24
11.2 通过 gitHub 平台实现本项目的全球域名	24
11.3 创建一个空的远程代码仓库	24
11.4 设置本地仓库和远程代码仓库的链接	25
参考文献:	27

基于 Web 客户端技术的个性化 UI 设计和实现

摘要: 由于互联网在近些年得到了高速发展, 具有跨操作系统这一优点的 Web 平台成为了广泛流行的软件开发技术, 本项目以 Web 客户端技术为研究学习对象, 广泛查阅了技术资料与相关文献, 尤其是 mozilla 组织的 MDN 社区的技术实践文章, 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。通过集成上述技术, 再应用本科的相关课程的知识, 实现了一个个性化的用户界面 (UI: uer interface) 的项目, 该用户界面以响应式技术为支撑做到了最佳适配用户屏幕, 程序可以动态适用于当前 PC 端和移动端设备; 在功能上以 DOM 技术和事件驱动模式的程序为支撑实现了对鼠标、触屏、键盘的底层事件响应和流畅支持, 为鼠标和触屏设计了一个对象模型, 用代码实现了对这类指向性设备的模拟 (这是本项目模型研究法的一次创新实践, 也是本项目的亮点。)。为了处理好设计和开发的关系, 项目用了工程思想管理, 使用了软件工程的增强式开发模式, 一共做了 6 次项目迭代开发, 每次迭代都包含软件的 4 个经典开发过程 (A:Analysis, D:Design, I: Implementation, T:Testing), 逐步编写了 UI 应用。为了分享和共享本代码, 与网上的开发者共同合作, 本项目还使用了 git 工具进行代码和开发过程日志记录, 一共做了 12 次提交代码的操作, 详细记录和展现了开发思路和代码优化的过程, 最后通过 gitbash 把项目上传到 github 上,

建立了自己的代码仓库，并将该代码仓库设置成为了 http 服务器，实现了本 UI 应用的全球便捷访问。

关键词：web；技术；设计；响应；UI

第 1 章 前言

1.1 学习计划

1.1.1 第一周

进行“三段论”样式的内容设计概要开发。通过分析需求和功能，初步构建应用程序的结构和逻辑。

1.1.2 第二周

实现网页窄屏代码，并进行了个性化 UI 设计中的鼠标模型 1.0 的研究，利用鼠标的监听事件，显示鼠标的实时坐标。

1.1.3 第三周

实现个性化 UI 设计中的鼠标模型 2.0，支持触屏功能。利用鼠标监听事件，实现页面元素拖动，并显示拖动距离。

1.1.4 第四周

实现个性化 UI 设计中的鼠标模型 1.0，利用 keydown 和 keyup 事件，显示按下键的字符以及 key 和 keyCode 值。

1.1.5 第五周

实现个性化 UI 设计中的鼠标模型 2.0，根据鼠标模型和用户的操作设计了 UI 的反馈程序。另外还增加了底层函数\$，用于抓取页面上的对象在代码中反复使用。

1.1.6 第六周

在页面主区域为移动端添加了触屏拖动事件，为移动端用户提供方便。并在键盘响应区添加了键盘的监听事件，监听键盘的按下和抬起

1.3 研究方法

为了确保本项目的科学性和实用性，本论文采取了六种研究方法。首先，我们采用文献综述法，查阅了大量关于 HTML5、CSS、JavaScript 以及相关技术的文献资料，了解了当前领域的发展动态和研究成果，为项目的技术使用和开发提供理论支持；使用原型设计法，使用工具绘制了界面原型，确保设计方案的可行性。其次，我们利用增量开发法，采

用迭代增量式开发模式，将整个项目划分为若干小模块，逐步开发和测试，且每次都对已有的功能进行优化，确保项目的质量的进度。最后，我们采用实验测试法和版本控制法，在浏览器的开发者面板中进行调试修改，并使用 Git 进行版本控制和代码管理，通过 GitHub 平台进行项目的托管和全球互联网部署。

第 2 章 Web 平台和客户端技术概述和文献综述

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。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，这也是我的毕设项目应用的技术路线 Web 是 World Wide Web（万维网）的简称，是指通过互联网进行交流和信息共享的一种方式。它是由一些互相链接的、使用标准化的 HTML 或 XML 等超文本格式文档组成的，这些文档通过超链接相互关联，用户可以通过浏览器访问并交互。

让我们先简单介绍一下 Web，也就是万维网的缩写。大多数人说“Web”而不是“World Wide Web”，我们将遵循这一惯例。网络是文档的集合，称为网页，由世界各地的计算机用户共享(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[2]。

Web 编程是一个很大的领域，不同类型的 Web 编程由不同的工具实现。所有的工具都使用核心语言 HTML，所以几乎所有的 web 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，所有的深度。这三种技术被认为是客户端 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^[3]。

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）；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。

第 3 章 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。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：

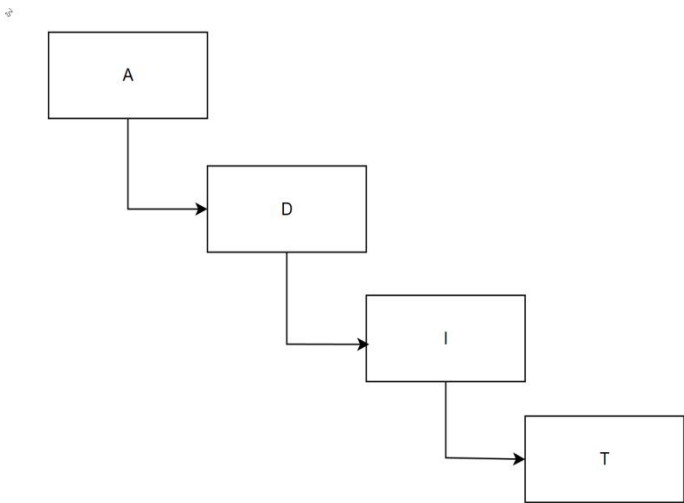


图 2-1 ADIT 过程

在增量模型中，软件是按一系列步骤开发的。开发人员首先完成整个系统的简化版本。这个版本代表整个系统，但不包括细节。图中显示了增量模型概念。

在第二个版本中，添加了更多的细节，而一些未完成，并再次测试系统。如果有问题，开发人员就会知道问题出在新功能上。在现有系统正常工作之前，它们不会添加更多的功能。这个过程一直持续到添加了所有需要的功能。

第 4 章 内容设计概要

4.1 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 4-1 用例图和 4-2Dom 树如图所示。

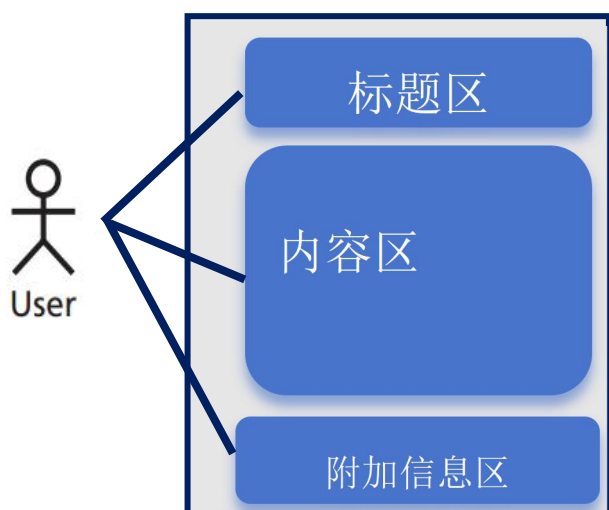


图 4-1 用例图

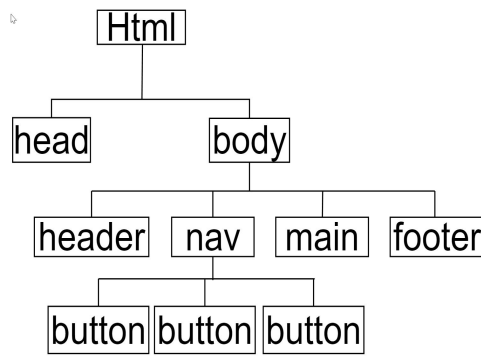


图 4-2Dom 树

4.2 项目的实现和编程

一、HTML 代码编写如下：

```
<header>
    《WebAppUI 设计开发论文》
</header>
<main>
    我的主题内容： ‘ ’
</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 ;
}
```

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 4-2 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 4-3 的二维码,运行测试本项目的第一次开发的阶段性效果。

《 我的毕设题目 》
我的主题内容: ‘读好书、练思维、勤编程’ @masterLijh 计算思维系列课程
CopyRight XXX 江西科技师范大学 2024-2025

图 4-2 PC 端运行效果图

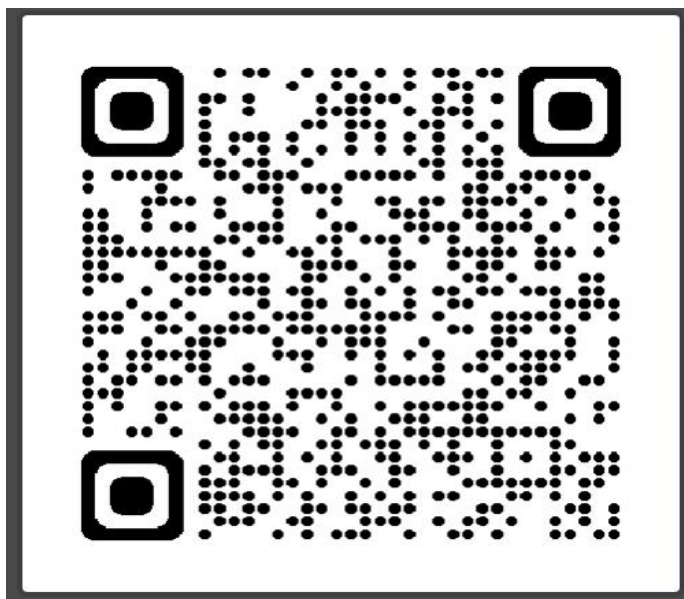


图 4-3 移动端二维码

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 江科师大刘鑫  
$ git config user.email marsterlijh@jxstnu.edu.cn  
$ 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：“三段论”式的内容设计概要开发  
[master (root-commit) 32de024] 项目第一版：“三段论”式的内容设计概要开发  
2 files changed, 46 insertions(+)  
create mode 100644 index.html  
create mode 100644 myCss.css
```

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：

```
$ git log
commit 32de0243f71ec4396f54175d83c805c999d473ca (HEAD -> master)
Author: 江科师大李健宏 <marsterlijh@jxstnu.edu.cn>
Date: Wed May 29 15:26:56 2024 +0800

项目第一版：“三段论”式的内容设计概要开发
```

第 5 章 移动互联时代的响应式设计和窄屏代码实现

响应式设计——适应显示硬件

计算机所使用的显示硬件千差万别，显示器的大小和分辨率取决于成本。设计师们选择让网页给出一般的布局准则，并允许浏览器选择如何在给定的计算机上显示页面，而不是为每种类型的显示提供每个网页的版本。因此，一个网页不能提供很多细节。例如，网页的作者可以指定一组句子组成一个段落，但作者不能指定诸如一行的确切长度或是否缩进段落开头等细节^[1]。

允许浏览器选择显示细节会产生一个有趣的结果:当通过两个浏览器或在硬件不同的两台计算机上浏览时，网页可能会显示不同的内容。如果一个屏幕比另一个屏幕宽，则可以显示的文本行的长度或图像的大小不同。重点是:网页给出了关于期望呈现的一般指导方针;浏览器在显示页面时选择详细信息。因此，同一网页在两台不同的计算机或不同的浏览器上显示时可能会略有不同^[1]。

用 JavaScript 开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。

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 。其用例图和 Dom 树如下图 5-1、5-2 所示，如代码块 4-1 所示：

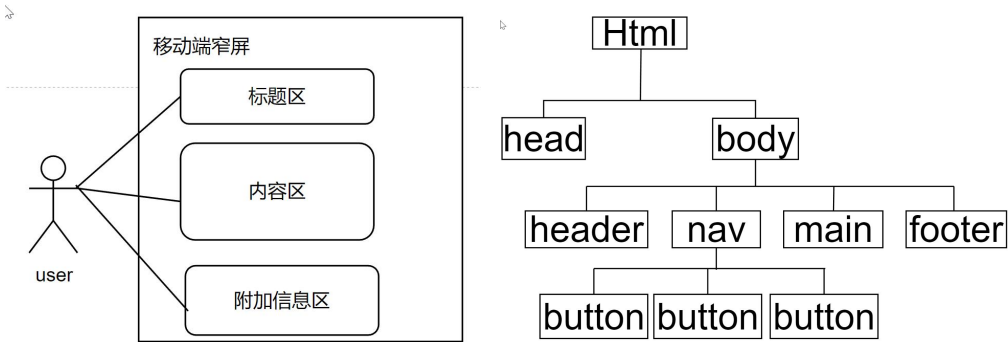


图 5-1 用例图

图 5-2 Dom 树

```

<style>
  *{
    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%;
  }
</style>

```

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

第 6 章 适用移动互联时代的响应式设计

与计算机一起使用的显示硬件种类繁多，显示器的大小和分辨率取决于成本。因此，设计师们选择让网页提供一般的布局指导方针，并让浏览器在给定的计算机上选择如何显示网页。因此，网页不会提供很多细节信息。例如，网页作者可以指定一组句子构成一段文本，但作者无法指定诸如每行的精确长度或是否缩进段落开头等细节信息^[1]。允许浏览器选择显示细节会产生一个有趣的结果:当通过两个浏览器或在硬件不同的两台计算机上浏览时，网页可能会显示不同的内容。如果一个屏幕比另一个屏幕宽，则可以显示的文本行的长度或图像的大小不同。重点是:网页给出了关于期望呈现的一般指导方针;浏览器在显示页面时选择详细信息。因此，同一网页在两台不同的计算机或不同的浏览器上显示时可能会略有不同^[1]。代码实施完成了下面几步：1. 通过把设备的高度全部分配给 body 对象，为软件四个区域做了百分比的高度比率的分配，实现各个区域的响应式设计；2. 为应用计算了基础字体的大小，并利用 body 的遗传机制，结合 css 字体的相对控制，实现了字体的响应式设计；具体内容展示如下图 1、2、3。其用例图和 Dom 树如下图 6-1、6-2 所示

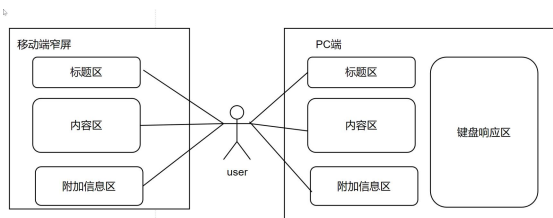


图 6-1 用例图

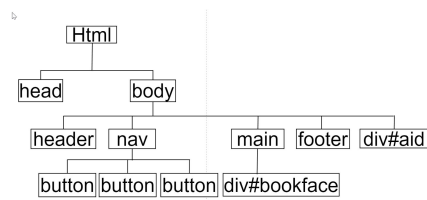


图 6-2 Dom 树

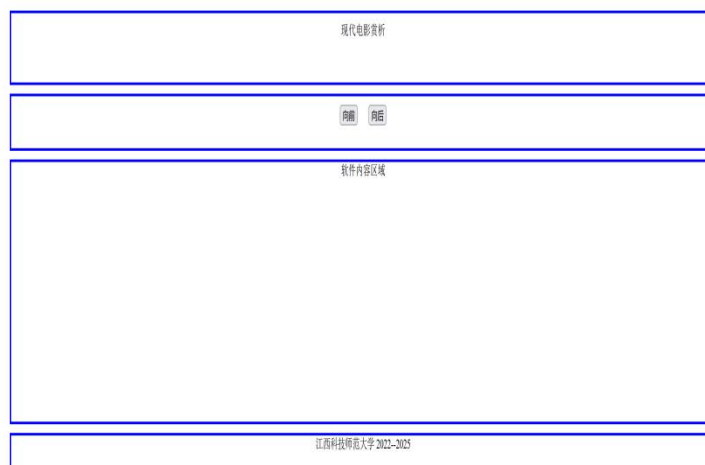


图 1 响应式设计和窄屏代码实现 PC 平台截图

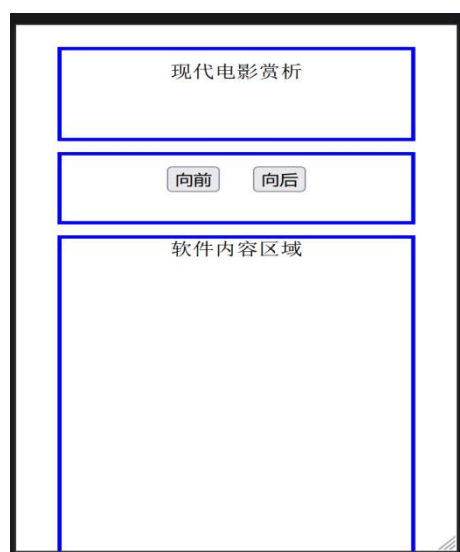


图 2 手机平台截图

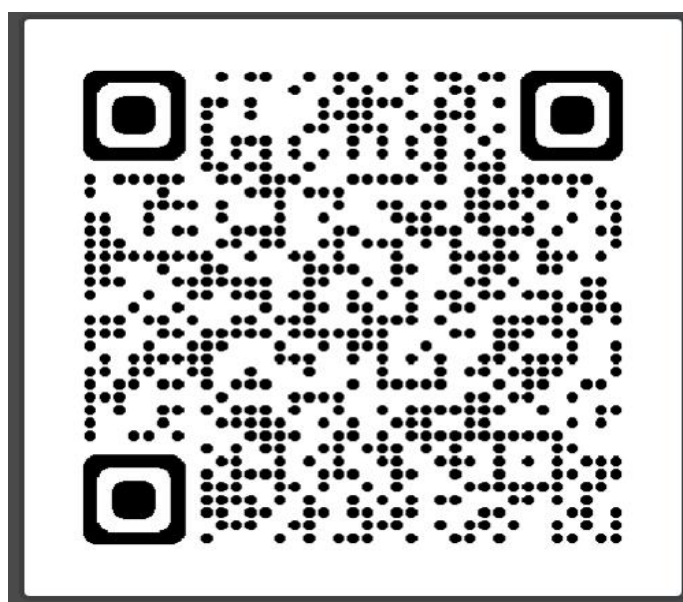


图 3 移动端二维码截图

第 7 章 个性化 UI 设计中鼠标模型

为了完成完成了鼠标对 UI 的初步控制,提交了两个文件 03.html 和 04.html。03.html 对屏幕唱过 1000px 的用户屏幕增加了用于键盘反馈的 ui 界面;还对主区域的书的封面元素,设定了鼠标按下和鼠标移动和抬起时间的响应。对 04.html 文件内增加了一个鼠标模型 mouse,用来模拟鼠标操作数据,并根据鼠标模型和用户的操作设计了 UI 的反馈程序。另外还增加了底层函数\$,用于抓取页面上的对象在代码中反复使用:

本次提交优化了 mouse 模型,通过研究鼠标的三个底层事件:mousedown、mousemove、一个可以拖动书的图案的 UI,该 UI 实现了软件介绍的书的切换 mouseup,设计了一个可以拖动书的图案的 UI,该 UI 实现了软件介绍的书的切换。Pc 端和手机平台的个性化 UI 设计中鼠标模型的截图展示如下图 1、2、3。其用例图和 Dom 树如下图 7-1、7-2 所示

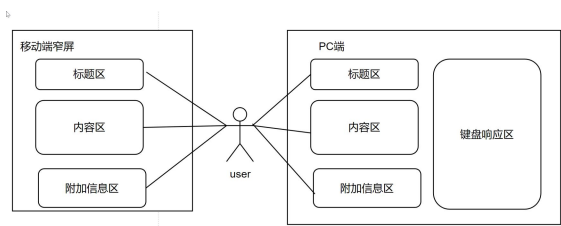


图 7-1 用例图

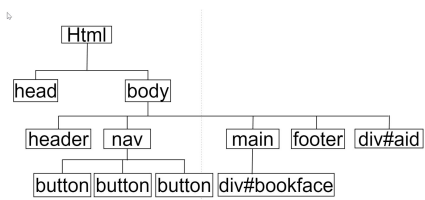


图 7-2 Dom 树

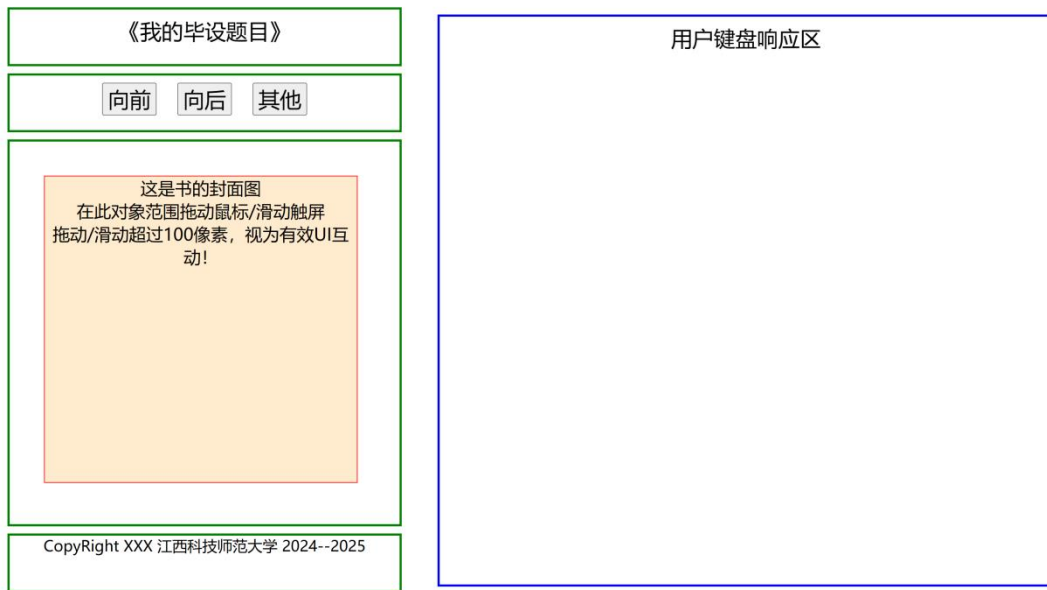


图 2 pc 平台个性化 UI 设计中鼠标模型



图 3 手机平台个性化 UI 设计中鼠标模型



图 4 移动端二维码

第 8 章 通用的 UI 设计

代码提交完成了触屏模型的模拟，当前代码设置对 main 元素(展示当前书的功能)有效。通过建立 Model.touch 模型，目前仅关注触摸对 x 轴方向的触摸，实现了不同书的切换。2、研究了基于 web 浏览器触屏的三个底层事件:touchstart、touchmove、touchend 结合 DOM 技术设计了一个可以滑动触屏控制书图案左右移动的 UI，该 UI 流畅地实现了书的切换:2、用户用触屏操作的动作细节包括:接触，移动，抬起，触的距离，通过逻辑综合判断这些因素，本片段代码设计了一个可行算法，判断了有效触摸，杜绝了无效触摸。

本次项目的设计还精简了 mouse 模型，通过研究鼠标的三个底层事件:mousedown、mousemove 一个可以拖动书的图案的 UI，该 UI 实现了软件介绍的书的切换;2、用户 mouseup，设计了用鼠标拖动书的动作细节很多，包括:按下，移动，抬起，移动距离，通过逻辑综合判断这些因素设计了一个可行算法，既判断了有效拖动(包括左右实现不同的切换方向)杜绝了无效拖动，也把拖动限定在半屏范围之内，展现了一个较为流畅可概念清晰的前后切换书的本例利用 CSS 动画开关设置，结合 1S 的异步代码，联合上面的 mouse 模型 GUI 模型，一个有动画效果的 UI。Pc 平台和手机平台的触屏和鼠标建模截图如下图 4、5、6。其用例图和 Dom 树如下图 8-1、8-2 所示。

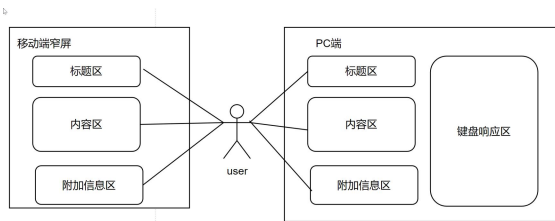


图 8-1 用例图

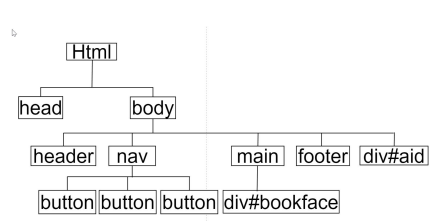


图 8-2 Dom 树

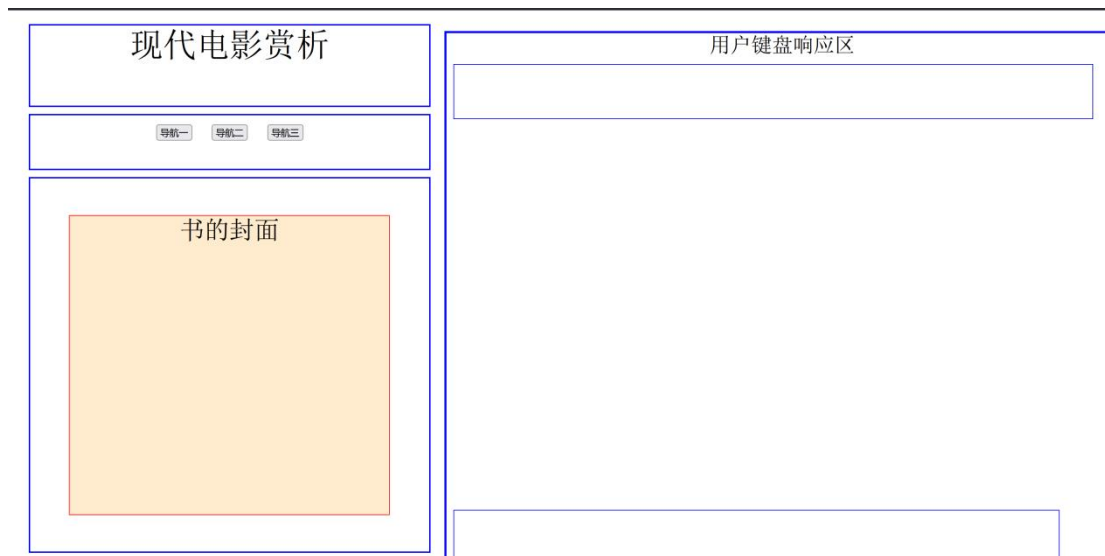


图 5 PC 平台触屏和鼠标建模



图 6 PC 平台触屏和鼠标建模



图 7 移动端二维码截图

第 9 章 UI 的个性化键盘控制

此次项目我们实现了用户输入的按键实时显示在“键盘响应区”，keydown 事件记录并输出按下键的值和键码，而 keyup 事件则在的 keyStatus 区域显示键已弹起的信息。此外，我们还会判断所抬起的键是否是字母或特殊符号，如果是，则在 typeText 区域进行

显示，如果不是，表示是功能键，我们选择不进行打印。

对于移动端设备，我们需要考虑到屏幕尺寸的影响。在窄屏模式下屏幕宽度小于 1000 像素时，由于空间限制，键盘响应区可能不会显示，此时我们需要考虑其他交互方式，如触摸事件。而在宽屏设备上由于有足够的空间，键盘响应区会正常显示，用户可以享受到与桌面端类似的键盘交互体验。

然而，优秀的设计不仅在于技术的应用，更在于对用户需求和习惯的理解。在实现自定义键盘控制时，我们必须确保其既高效又直观，避免给用户带来困扰。例如，虽然可以阻止 Enter 键的默认提交功能，但大多数情况下，用户期待 Enter 键在表单中起到提交的作用，因此我们需要在适当的情况下谨慎使用 `event.preventDefault()`。通过不断实践和优化，我们可以根据不同的用户群体和场景调整键盘控制的实现。无论是桌面端的复杂交互，如游戏控制、表单验证，还是移动端的简洁操作，`keydown` 和 `keyup` 事件结合 `event.preventDefault()` 都能提供强大的支持。这些底层事件为我们打开了新的设计思路，让我们能够创造出前所未有的用户体验。

总之，`keydown` 和 `keyup` 事件结合 `event.preventDefault()` 为 Web 开发者提供了丰富的工具，让我们能够在交互设计中实现更多可能性。通过深思熟虑的设计和不断迭代，我们可以不断拓展 Web 应用的交互边界，为用户提供更加自然、流畅的使用体验，从而推动 Web 开发领域的创新和发展。其用例图和 Dom 树如下图 9-1、9-2 所示

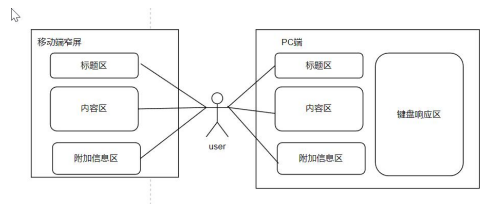


图 9-1 用例图

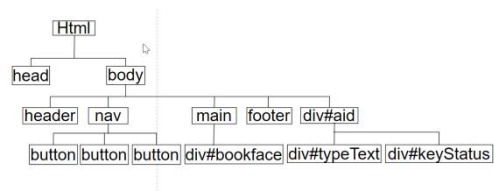


图 9-2 Dom 树

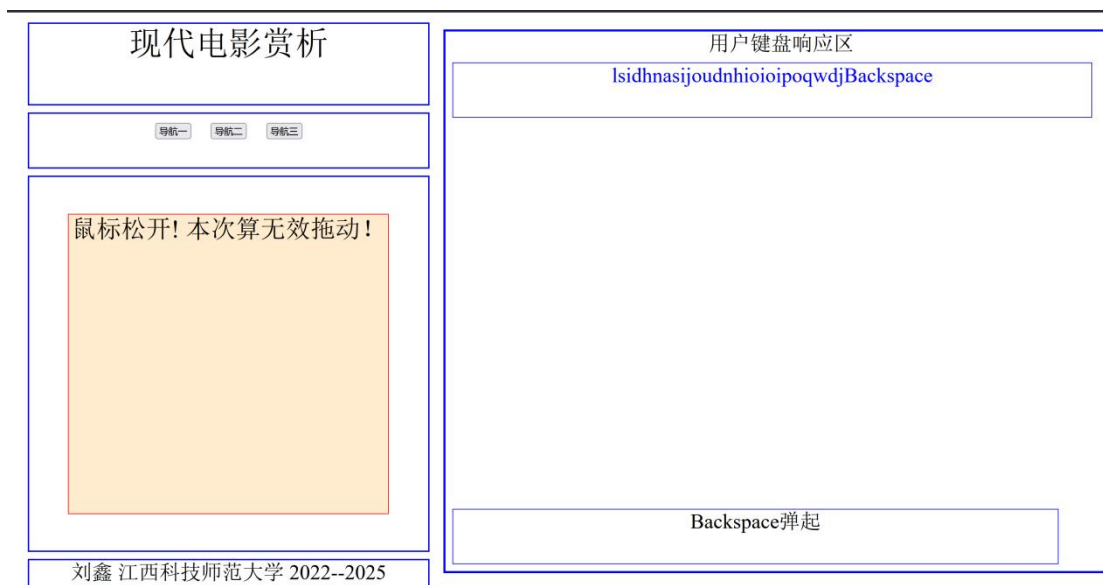


图 8 PC 平台 UI 的个性化键盘控制

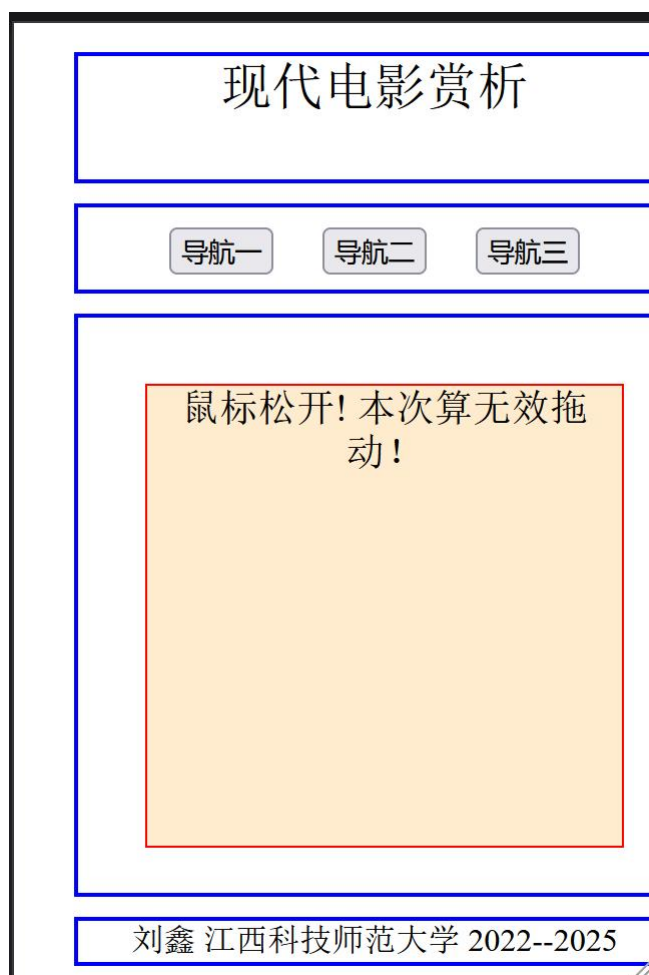


图 9 手机 PC 平台 UI 的个性化键盘控制平台 UI 的个性化键盘控制

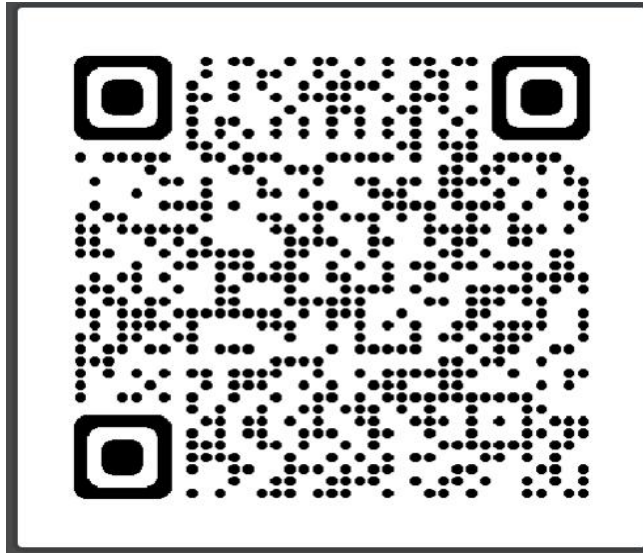


图 10 移动端二维码截图

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 ;  
    }  
}
```

```

    let puncs =
['~','`','!','@','#','$','%','^','&','*','(',')','-','_','+','=',' ','.',';',':','<','>','?','/',' ','\','\"'] ;
    if ( (k >= 'a' && k <= 'z') || (k >= 'A' && k <= 'Z')
|| (k >= '0' && k <= '9')) {
        console.log("letters") ;
        return true ;
    }
    for (let p of puncs ){
        if (p === k) {
            console.log("puncs") ;
            return true ;
        }
    }
    return false ;
    //提出更高阶的问题，如何处理连续空格和制表键 tab?
} //function printLetter(k)
});

```

第 10 章 本项目中的高质量代码

这是一本关于计算机教学的书。今天，电脑就像螺丝刀一样常见，但它们要复杂得多，让它们做你想让它们做的事情并不总是那么容易。

如果你给你的电脑的任务是一个常见的，很容易理解的任务，比如显示你的电子邮件或像计算器一样工作，你可以打开适当的应用程序并开始工作。但对于独特的或开放式的任务，可能没有应用程序。

这就是编程可能发挥作用的地方。编程是构建程序的行为——一组告诉计算机该做什么的精确指令。因为计算机是愚蠢的、迂腐的野兽，编程从根本上来说是乏味和令人沮丧的。幸运的是，如果你能克服这个事实，甚至享受用愚蠢的机器可以处理的方式思考的严谨性，编程是有回报的。它可以让你在几秒钟内完成手工永远做不完的事情。它是一种让你的计算机工具做它以前不能做的事情的方法。它还提供了一个很好的抽象思维练习。[6] 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。

```
{ //Code Block begin
```

```

const ORIGINAL_LEFT = 8 + '%';
const VALID_DRAG_DISTANCE = 100;

let handleBegin = function (ev) {
  Pointer.isDown = true;
  // 记录下起始位置
  Pointer.x = ev.pageX || ev.touches[0].pageX
  Pointer.y = ev.pageY || ev.touches[0].pageY
  if (ev.touches) {
    console.log("Touch begin : " + "(" + Pointer.x + "," + Pointer.y
+ ")");
    $("bookface").textContent = "触屏事件开始, 坐标: " + "(" +
Pointer.x + "," + Pointer.y + ")";
  } else {
    console.log("MouseDown at x: " + "(" + Pointer.x + "," + Pointer.y
+ ")");
    $("bookface").textContent = "鼠标按下, 坐标: " + "(" + Pointer.x
+ "," + Pointer.y + ")";
  }
};

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

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

```

```

        if (Pointer.isDown) {
            if (ev.touches) {
                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 {
                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';
            }
        }
    };
    $("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);
    $("bookface").addEventListener('click', function (ev) {
        console.log('本次触发了点击事件: click。');
    });
    /* $("body").addEventListener("keypress", function (ev) {
        let key = ev.key;
        $("outputText").textContent += key;
    }); */
    // keydown 和 keyup 事件增加 ev.preventDefault()以后, 键盘 keypress
事件被阻止
    $("body").addEventListener("keydown", function(ev) {
        ev.preventDefault();
        let key = ev.key;
        $("keyStatus").textContent = '按下键: ' + key;
    });
    $("body").addEventListener("keyup", function(ev) {
        ev.preventDefault();
        let key = ev.key;
        $("keyStatus").textContent = key + "弹起";
        $("typeText").textContent += key;
    });
} //Code Block end

```


第 11 章 用 gitBash 工具管理本项目的 http 服务器

11.1 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来自 GNU 项目的 shell 程序，名为 bash。这个名字是 Bourne -again shell 的首字母缩略词，指的是 bash 是 sh 的增强替代品，sh 是 Steve Bourne 编写的原始 Unix shell 程序。

像 Windows 一样，像 Linux 这样的类 unix 操作系统用所谓的分层目录结构来组织文件。这意味着它们被组织成树状的目录模式(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，子目录包含更多的文件和子目录，以此类推。

11.2 通过 gitHub 平台实现本项目的全球域名

11.3 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 *

 masterLijh ▾

Repository name *

/

✓ userName.github.io is available.

Great repository names are short and memorable. Need inspiration? How about [expert-rotary-phone](#) ?

Create repository

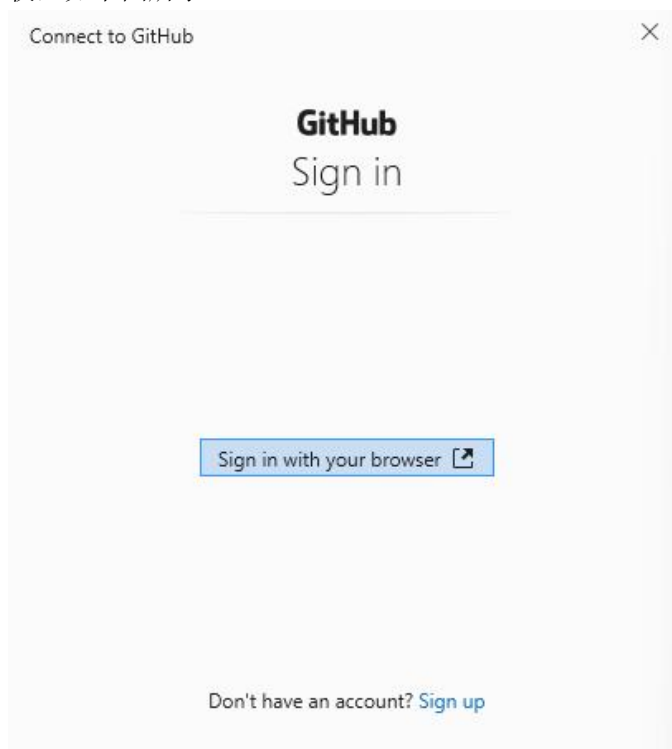
点击窗口右下角的绿色“Create repository”，则可创建一个空的远程代码仓库。

11.4 设置本地仓库和远程代码仓库的链接

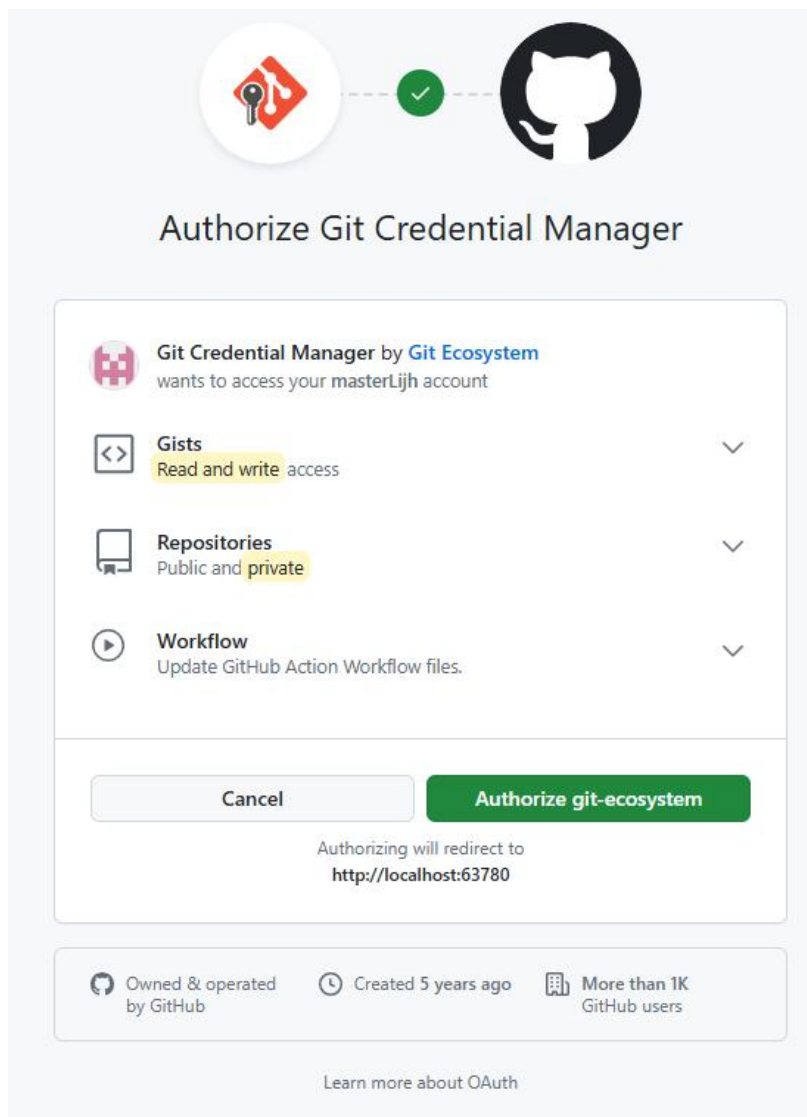
进入本地 webUI 项目的文件夹后，通过下面的命令把本地代码仓库与远程建立密钥链接

```
$ echo "WebUI 应用的远程 http 服务器设置" >> README.md
$ git init
$ git add README.md
$ git commit -m "这是我第二次把代码仓库上传至 gitHub 平台"
$ git branch -M main
$ git remote add origin
    liziyutou/liuxin: 2024/4/25 \(github.com\)
$ git push -u origin main
```

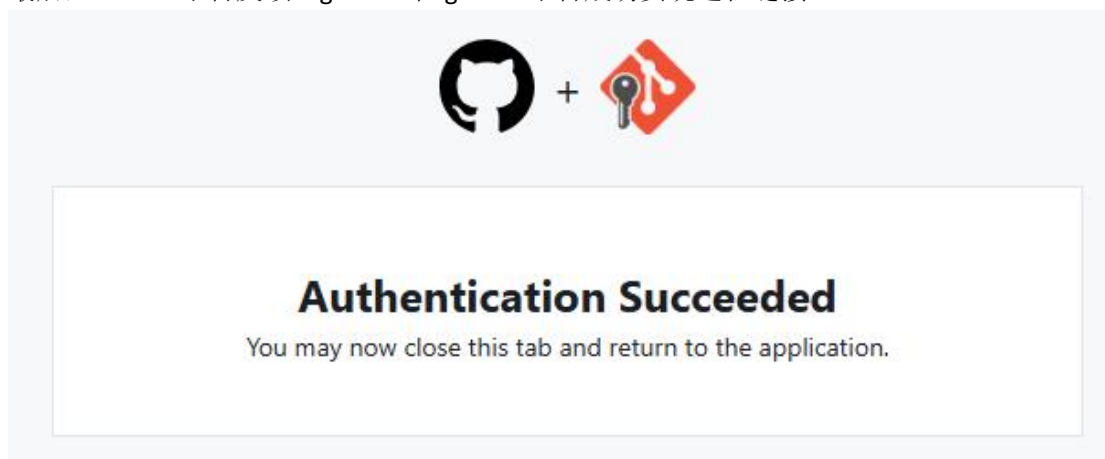
本项目使用 window 平台，gitbash 通过默认浏览器实现密钥生成和记录，第一次链接会要求开发者授权，如下图所示：



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。

参考文献：

- [1] . W3C. W3C's history. W3C Community. [EB/OL]. <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
- [8] . 王政.Web 前端开发技术以及优化研究[J].电脑知识与技术,2013,9(22):5037-5038.
- [9] . 彭显雯. 基于 MVVM 模式的响应式轻量级前端组件设计与实现[D].华中科技大学,2019.
- [10] . 仇礼钦,王鑫,盛飞龙,等.基于 Git 的软件项目管理配置方法及应用实践[J].机电工程技术,2023,52(05):223-227.