

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 的设计与实现

（外文）： Design and implementation of personalized UI 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： 李晨

学 号： 20213635

指导教师： 李健宏

2024 年 6 月 27 日

目 录

基于 Web 客户端技术的个性化 UI 的设计和实现	III
(Customized UI design and Programming based on Web client technology)	III
Abstract	IV
1. 前言	1
1.1 毕设任务分析	1
1.2 研学计划	1
1.3 研究方法	2
2. 技术总结和文献综述	3
2.1 Web 平台和客户端技术概述	3
2.2 项目的增量式迭代开发模式	4
3. 内容设计概要	4
3.1 分析和设计	4
3.2 项目的实现和编程	5
3.3 项目的运行和测试	6
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	16
6.1	分析和设计	16
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	18
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	19
7.3	项目的运行和测试	21
7.4	项目的代码提交和版本管理	22
8.	UI 的个性化键盘交互控制的设计开发	23
8.1	分析和设计	23
8.2	项目的实现和编程	23
8.3	项目的运行和测试	25
8.4	项目的代码提交和版本管理	25

9. 谈谈本项目中的高质量代码	26
10. 用 gitBash 工具管理项目代码仓库和 http 服务器	27
10.1 经典 Bash 工具介绍	27
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:	33
15. 写作指导:	错误! 未定义书签。

基于 Web 客户端技术的个性化 UI 的设计和实现

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

科师大元宇宙产业学院 2021 级 李晨

摘要: 近十年来, html5 为核心的 web 标准的软件开发技术以其跨平台、开源的优势广泛地运用在各个领域的应用软件开发中。通过分析本次毕设任务, 本项目选择 html5 的 web 客户端技术为技术路线, 展开对程序设计和软件开发的研究和实践。通过广泛查阅相关技术书籍、开发者论坛和文献, 设计开发了一个个性化的用户界面 (UI) 的应用程序。在开发中综合应用了 html 语言进行内容建模、css 语言展开 UI 的外观设计、javascript 语言编程实现 UI 的交互功能, 除直接使用了 web 客户端最底层的 API 外, 本项目的每条代码都是手工逐条编写, 没有导入他人的任何的代码 (框架和库)。本项目也采用了响应式设计编程, 可以智能地适应移动互联网时代用户屏幕多样化的需要; 另外大量地运用了面向对象的程序设计思想, 比如用代码构建了一个通用的 pointer 模型, 该模型仅用一套代码就实现了对鼠标和触屏的控制, 实现了高质量的代码, 这也是本项目的亮点。从工程管理的角度看, 本项目采用的增量式开发模式, 以逐步求精的方式展开了六次代码的增量式重构 (A:Analysis, D:Design, I: Implementation, T:Testing), 比较愉快地实现项目的设计开发和测试。从代码的开源和分享的角度看, 本项目采用了 git 工具进行版本管理, 在漫长的开发

过程中重构代码六次并正式做了代码提交，另外在测试中修改提交了代码两次，最后利用 gitbash 工具 把本项目的代码仓库上传到著名的 github 上，再利用 github 提供的 http 服务器，本项目实现了 UI 应用在全球互联网的部署，我们可以通过地址和二维码便捷地跨平台高效访问这个程序^[1]。

关键词：计算机；web；html；git；应用软件开发；

Abstract

In recent ten years, html5 as the core of the web standard software development technology with its cross-platform, open source advantages widely used in various fields of application software development. Through the analysis of the completion task, the project chooses html5 web client technology as the technical route, and carries out the research and practice of program design and software development. Through extensive access to technical books, developer forums, and literature, a personalized user interface (UI) application was designed and developed. In the development, html language was comprehensively applied to conduct content modeling, css language to develop UI appearance design, and javascript language programming to realize UI interaction functions. In addition to directly using the API at the lowest level of the web client, each code of the project was written manually one by one, without importing any code (framework and library) of others. This project also adopts responsive design and programming, which can intelligently adapt to the diversified needs of users' screens in the era of mobile Internet; In addition, a large number of object-oriented programming ideas are used, such as using code to build a general pointer model, which only uses a set of code to achieve the control of the mouse and touch screen, to achieve high quality code, which is also the highlight of this project. From the perspective of engineering management, the incremental development mode adopted in this project has carried out six incremental code reconstructions (A:Analysis, D:Design, I: Implementation, T:Testing) in a way of gradual refinement, thus achieving the design, development and testing of the project happily. From the perspective of open source and sharing of code, this project adopts git tool for version management, rebuilds the code six times in the long development process and formally submits the code, in addition to modifying and submitting the

code twice in the test. Finally, gitbash tool is used to upload the code warehouse of this project to the famous github. Using the http server provided by github, this project realizes the deployment of UI application on the global Internet, and we can conveniently and efficiently access this program across platforms through the address and two-dimensional code.

Key words: computer; web; html; git; Application software development

1. 前言

1.1 毕设任务分析

为了完成毕业设计任务，我们首先需要深入探讨个性化用户界面(UI)的概念及其在网络客户端技术中的应用，这有助于我们理解其在现实世界中的重要性和应用场景。紧接着，我们将开展广泛的技术调研，涵盖前端技术如 HTML、CSS 和 JavaScript，以及后端技术包括 Node.js、Python 的 Flask 或 Django 框架，并对数据库和个性化算法进行深入了解^[2]。

在确立了技术基础后，我们将对项目需求进行详细分析，包括用户信息管理、偏好设置分析和定制化内容展示等方面。随后，我们将设计系统的总体架构，并根据这一架构进行用户界面的规划，以确保满足用户体验和个性化需求。

在设计阶段完成后，我们将进入开发阶段，包括前端界面的构建、后端服务的搭建以及数据库的设置。开发完成后，我们将进行系统的测试和优化，确保系统的稳定性和性能。

最终，我们将撰写毕业设计文档，并在整个设计和实现过程中进行评估和反思，提出进一步的改进建议，以确保毕业设计任务的圆满完成。

1.2 研学计划

本研究计划致力于探究基于网络客户端技术的定制化用户界面 (UI) 的设计和开发，目的是适应用户的个性化需求和偏好。通过文献回顾和技术研究，我将对前端开发技术、后端服务架构以及定制化算法进行深入分析。在系统设计阶段，我将构建一个原型系统，该系统能够根据用户的偏好动态调整界面布局，并在开发阶段实现前端界面、后端服务的建立以及定制化算法的整合。

在评估阶段，我将对系统的性能和用户满意度进行测试，并基于测试结果提出优化建议。通过这项研究，我的目标是不仅深化对定制化 UI 设计和实现策略的理解，而且为增强网络应用的用户交互体验和市场竞争力贡献实质性的见解^[3]。

1.3 研究方法

本研究计划旨在全面探索基于 Web 客户端技术的个性化用户界面（UI）的设计与实现，以满足用户多样化的需求和偏好。首先，通过文献综述，我们将通过学术文献数据库如 IEEE Xplore 和 Google 学术，以及相关领域的期刊会议论文，对个性化 UI 的现有研究进行全面梳理，涵盖其定义、发展历程、关键技术与应用场景。接着，进行技术调研，深入了解前端技术（HTML、CSS、JavaScript 等）、后端服务架构（Node.js、Python Flask/Django 等）以及个性化算法（如协同过滤、内容推荐等），确保对所需技术有全面的认识^[4]。

在系统设计阶段，我们将确定个性化 UI 系统的整体架构，关注其可扩展性、灵活性和性能，同时设计用户偏好数据的收集、存储和管理方案，确保数据安全和隐私保护。系统实现阶段，将使用 HTML、CSS 和 JavaScript 等前端技术，开发用户界面，包括页面布局、交互设计和视觉呈现。后端服务将使用 Node.js、Python Flask/Django 等框架搭建，实现用户数据的存储和管理。此外，将个性化算法集成到系统中，以实现根据用户偏好动态调整 UI 的功能。

系统评估阶段，将对系统性能进行评估，包括页面加载速度和响应时间，确保系统的稳定性和高效性。通过用户调查和行为分析，评估用户体验，发现并解决潜在问题。基于评估结果，提出改进建议，包括优化算法和界面设计，以提升系统性能和用户体验。同时，进行系统优化，包括代码优化和性能调优，提高系统稳定性和响应速度。

最后，撰写研究论文，介绍个性化 UI 设计与实现的方法、技术和实验结果，并总结研究成果和经验，提出未来工作的展望和建议。通过这一研究方法，我们期望为提升 Web 应用的用户体验和竞争力做出实质性的贡献。

2. 技术总结和文献综述

2.1 Web 平台和客户端技术概述

Web 之父 Tim Berners-Lee 在发明 Web 的基本技术架构以后,就成立了 W3C 组织,该组织在 2010 年后推出的 HTML5 国际标准,结合欧洲 ECMA 组织维护的 ECMAScript 国际标准,几乎完美缔造了全球开发者实现开发平台统一的理想,直到今天,科学家与 Web 行业也还一直在致力于完善这个伟大而光荣的理想^[1]。学习 Web 标准和 Web 技术,学习编写 Web 程序和应用有关工具,最终架构一套高质量代码的跨平台运行的应用,是我的毕设项目应用的技术路线。

Web 平台技术是构建个性化用户界面 (UI) 的基础,其中 HTML (超文本标记语言) 负责定义网页的结构和内容,允许开发者创建文本、图像和链接等网页元素。CSS (层叠样式表) 则用于控制网页的样式和布局,通过字体、颜色、大小和位置等属性对 HTML 元素进行美化和布局设计。JavaScript, 一种基于对象和事件驱动的脚本语言,为网页增添了动态效果、交互功能和数据处理能力,极大地丰富了用户的使用体验。

前端框架和库,如 React.js、Vue.js 和 Angular,提供了高效的开发模式,简化了开发流程,同时提高了代码的可维护性和可扩展性。客户端技术,尤其是浏览器,作为 Web 应用的直接交互界面,允许用户访问和显示网页内容。随着移动互联网的兴起,移动端技术,包括移动端浏览器、混合应用 (Hybrid App) 和原生应用 (Native App),成为 Web 应用开发的重要方面。响应式设计通过 CSS3 的媒体查询和弹性布局等技术,确保网页能够自适应不同设备的分辨率和屏幕尺寸,提供一致的良好用户体验。

跨平台兼容性是 Web 开发中的一个关键考虑因素,开发者需要确保 Web 应用在不同平台和设备上的兼容性和稳定性,这涉及到对各种浏览器的兼容性、移动端和桌面端的差异等因素的考量,以及在开发过程中进行充分的测试和调试。

综合这些 Web 平台和客户端技术，开发者能够实现网页的结构、样式和交互效果，为用户提供个性化且友好的用户界面。

2.2 项目的增量式迭代开发模式

增量式迭代开发是一种高效的软件开发策略，它通过将整个项目拆分为多个小的、可交付的增量或迭代来实现。每个增量都是一个完整的功能子集，能够独立部署并为用户带来即时价值。这种方法的实施步骤包括需求分析与规划，其中项目目标和范围被明确，用户需求被分析，功能和特性被定义，以及增量迭代计划被制定。设计阶段则依据迭代需求进行系统设计和架构规划，并确定技术选型与开发工具。开发与测试阶段涉及功能开发和一系列测试，如单元测试、集成测试和系统测试，以确保代码质量和功能完整性。部署与发布阶段则将增量部署到测试或生产环境，并进行用户验收测试。反馈与优化阶段收集用户反馈，根据需求变更调整后续迭代计划。迭代循环确保项目功能持续完善和扩展。

增量式迭代开发的特点包括快速交付，每个增量都能迅速提供给用户；灵活性和适应性，允许根据用户反馈和需求变化灵活调整计划；降低风险，通过分阶段开发和测试减少项目风险；持续改进，通过不断收集用户反馈优化项目；以及增量累积，每个新增量都建立在前一个基础上，逐步构建完整的项目功能。这种开发模式不仅提高了开发效率，还确保了项目能够适应用户不断变化的需求和期望，为项目团队提供了一种高效、灵活且用户中心的开发策略。

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 4-1 用例图所示：

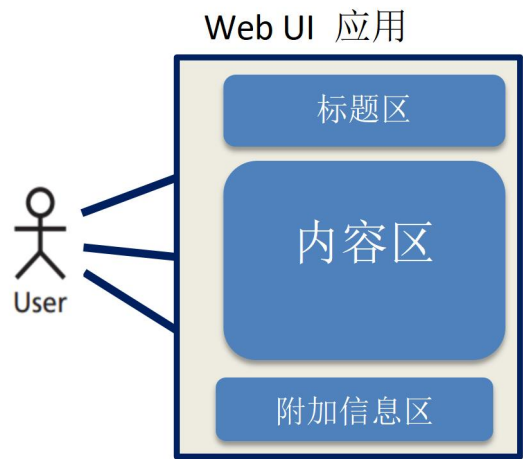


图 4-1 用例图

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 李晨 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。

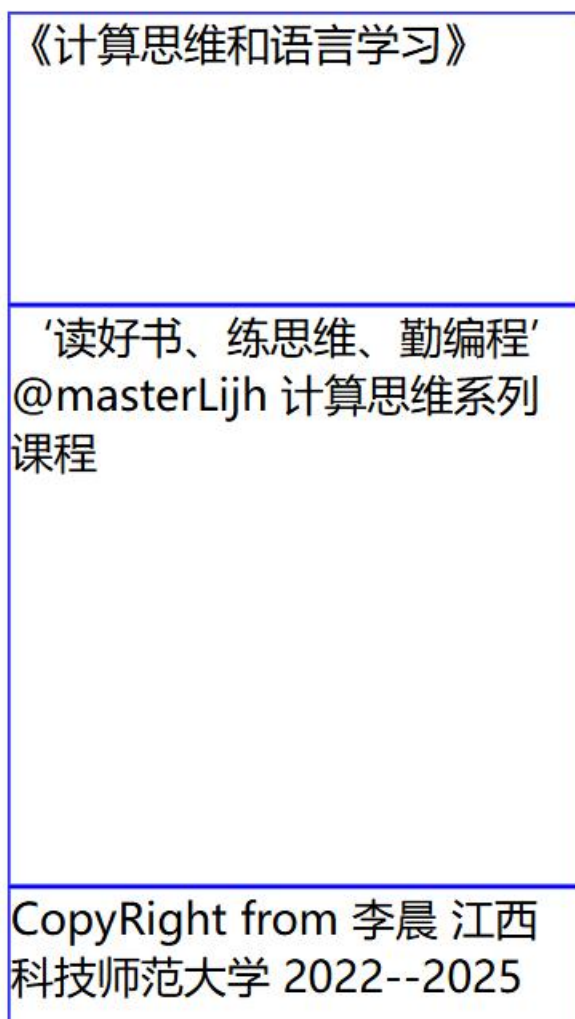


图 4-2 PC 端运行效果图

扫描 QR 码



<https://diamondful.github.io/exp/1.1.html>

图 4-3 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 diamondful
$ git config user.email 3207307701@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
```

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

移动互联时代的 UI 开发必须考虑到用户在各种设备上的使用体验，尤其是针对窄屏终端的响应式设计。窄屏终端通常指的是智能手机和平板电脑等小尺寸屏幕设备，设计师需要特别关注这些设备上的布局 and 交互方式，以确保用户能够方便地访问和操作应用程序或网站用例图和 dom 图如下图 4.1。

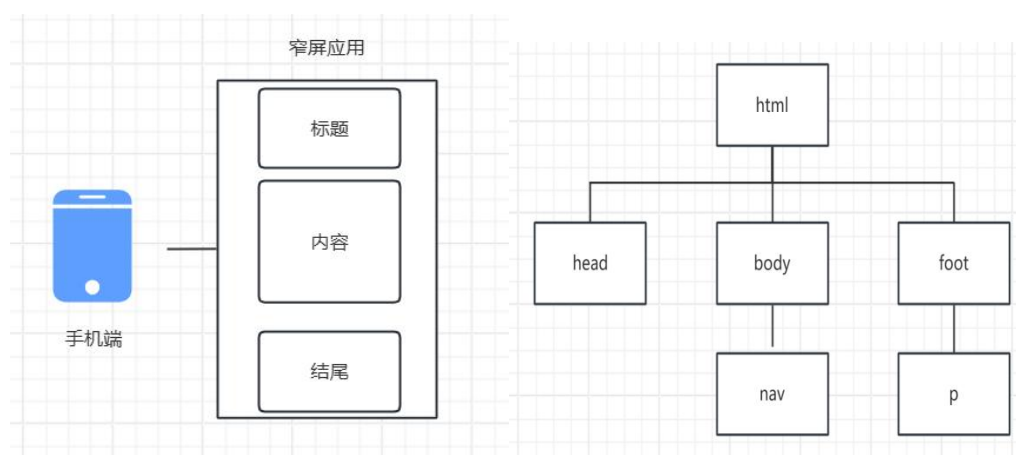


图 4.1

4.2 项目的实现和编程

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：

```
<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

4.3 项目的运行和测试

本文此处给出 PC 端用 Chrome 浏览器打开项目的结果，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二维码，运行测试本项目的第一次开发的阶段性效果。



图 4.3

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，我们的目标是提交本次版本代码并增加注释。

进入 gitBash 命令行后，按次序输入以下命令：

```
cd /d  
Cd /abc  
Git add exp/1.2.html  
Git commit -m "第四章版本"
```

截图图下所示：

```
Author: diamondful <3207307701@qq.com>  
Date:   Wed Jun 5 13:56:07 2024 +0800
```

增加全部文件

```
commit 051a77b4e8cab914c7b6c16eac9966e7e72787b0  
Author: diamondful <3207307701@qq.com>  
Date:   Wed Jun 5 13:46:45 2024 +0800
```

第四个版本

```
commit 6e84f66ceaea74de4bce9ac33b31d027e6aff8c6  
Author: diamondful <3207307701@qq.com>  
Date:   Wed Apr 24 09:36:48 2024 +0800
```

本次代码进行了响应式设计

```
commit 14d33a5052f467e03d92290dee63e740a1d70770  
Author: diamondful <3207307701@qq.com>  
Date:   Tue Apr 23 09:23:42 2024 +0800
```

```
:|
```

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

经过了上一章对响应式技术开发适配移动端设备的网页的设计经验后，这一章我们通过更高级的代码来实现更多的功能，阐述移动互联时代的用户终端的多样性，以及如何用 `css` 语言和 `JavaScript` 语言实现响应式设计，用例图和 `dom` 图如下图 5.1^[5]。

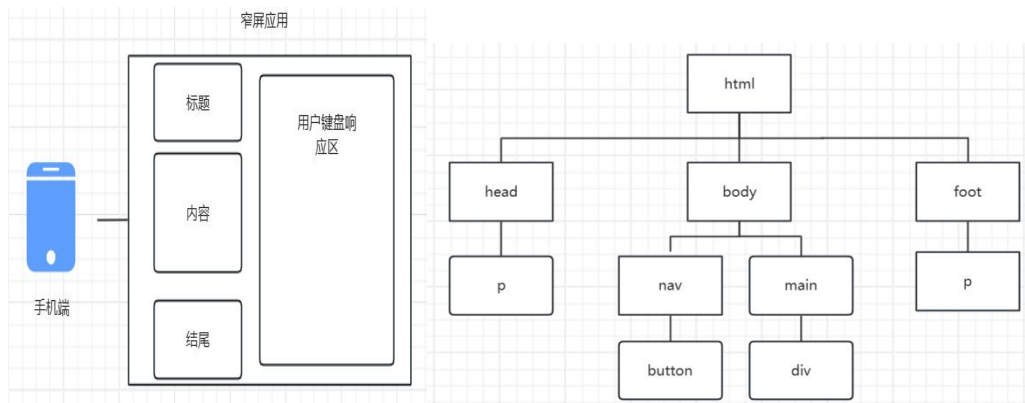


图 5.1

5.2 项目的实现和编程

页面的整体布局如代码块 5.2 所示

```

<header>
  <p id="book">
    《现代电影赏析》
  </p>
</header>
<nav>
  <button>向前</button>
  <button>向后</button>
  <button>暂停</button>
</nav>

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

<footer>

  Copyright from 李晨 江西科技师范大学 2022--2025

</footer>
  
```

代码块 5.2

其次，在本次代码中，首次尝试了对鼠标的设计 UI 控制，如下代码块 5.3。

//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="离开";
});

$("#body").addEventListener("keypress",function(ev){
    let k=ev.key;
    let c=ev.keyCode;
    let s1="代码是: ";
    let s2="按键是: ";
    $("#keyboard").textContent=s2+k+"\n"+s1+c;
});
```

代码块 5.3

5.3 项目的运行和测试

本文此处给出 PC 端用 Chrome 浏览器打开项目的结果，如下图 5-4 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二维码，运行测试本项目的本次开发的阶段性效果。



图 5.4

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，我们的目标是将本次代码上传到 github 仓库，并进行本次提交的信息备注。

进入 gitBash 命令行后，按次序输入以下命令：

```
cd /d
Cd /abc
Git add exp/1.3.html
Git commit -m "增加全部文件"
```

截图图下所示：

```
Author: diamondful <3207307701@qq.com>
Date:   Wed Jun 5 13:56:07 2024 +0800

    增加全部文件
```

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

经过了上一章对响应式技术开发适配移动端设备的网页的设计经验后，这一章我们通过更高级的代码来实现对鼠标交互功能的初步开发，阐述移动互联时代的用户终端的多样，用例图和 dom 图如下图 6.1。

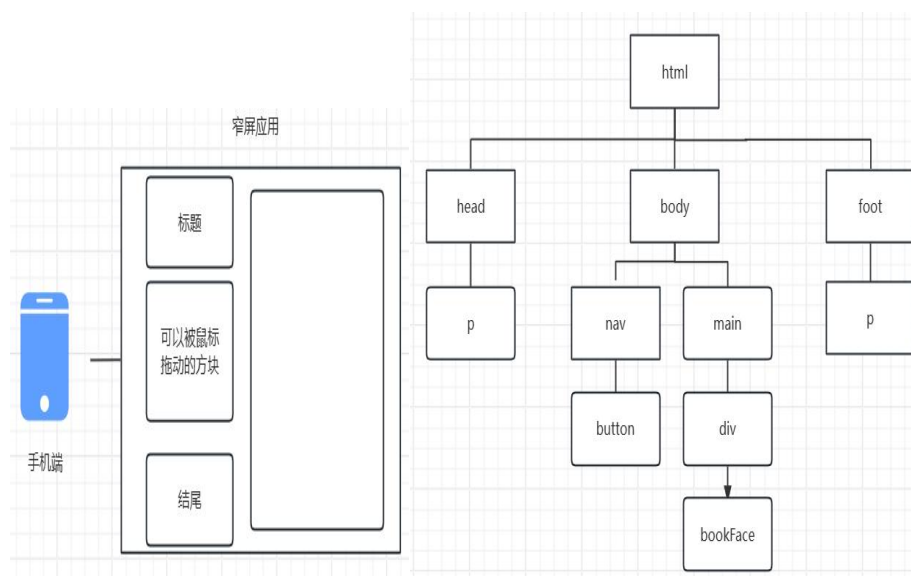


图 6.1

6.2 项目的实现和编程

主要实现功能的代码块如下代码块 6.2.

```
//尝试对鼠标设计 UI 控制
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' ;
    }
}

```

代码块 6.2

6.3 项目的运行和测试

本文此处给出 PC 端用 Chrome 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二维码，运行测试本项目的本次开发的阶段性效果。



图 6.3

6.4 项目的代码提交和版本管理

```
cd /d
Cd /abc
Git add exp/1.4.html
Git commit -m "提交关于鼠标交互的设计开发版本"
```

截图图下所示：

```
diamond@LAPTOP-O9TI1BHJ MINGW64 /d/abc/exp (main)
$ git commit -m "提交关于鼠标交互的设计开发版本"
[main 9b709a0] 提交关于鼠标交互的设计开发版本
1 file changed, 2 insertions(+), 2 deletions(-)

diamond@LAPTOP-O9TI1BHJ MINGW64 /d/abc/exp (main)
$ git log
commit 9b709a02311aab9934232a712d8b9a9321cf71b4 (HEAD -> main)
Author: diamondful <3207307701@qq.com>
Date: Wed Jun 12 10:27:18 2024 +0800
```

提交关于鼠标交互的设计开发版本

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

经过了上一章对鼠标交互设计网页的设计经验后，这一章我们通过更高级的代码来实现对触屏和鼠标功能的通用交互操作初步开发，阐述移动互联时代的用户终端的多样，用例图和 dom 图如下图 7.1。

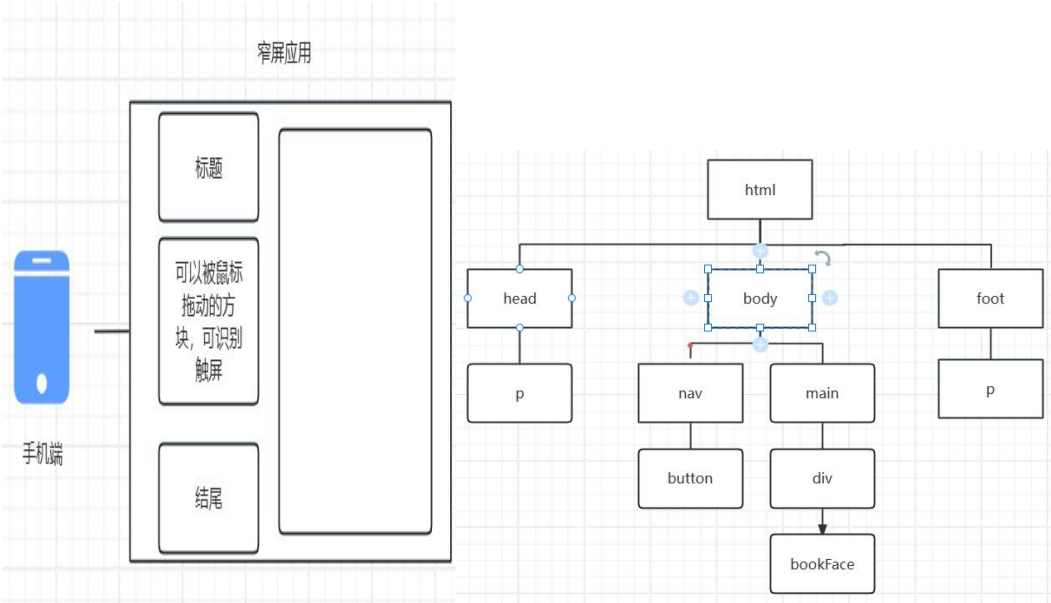


图 7.1

7.2 项目的实现和编程

主要实现功能的代码块如下代码块 7.2.

```
//尝试对鼠标和触屏设计一套代码实现 UI 控制
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+")";
    }
};
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%' ;
        }
    }
};
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' ;
        }
    }
};

$("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 ;
});

```

代码块 7.2

7.3 项目的运行和测试

本文此处给出 PC 端用 Chrome 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二维码，运行测试本项目的本次开发的阶段性效果。



图 7.3

7.4 项目的代码提交和版本管理

```
cd /d
Cd /abc
Git add exp/1.5.html
Git commit -m "提交对触屏鼠标的通用交互操作版本"
```

截图图下所示：

```
diamond@LAPTOP-O9TI1BHJ MINGW64 /d/abc/exp (main)
$ git add 1.5.html

diamond@LAPTOP-O9TI1BHJ MINGW64 /d/abc/exp (main)
$ git commit -m "提交对触屏鼠标的通用交互操作版本"
[main 631d0bf] 提交对触屏鼠标的通用交互操作版本
1 file changed, 1 insertion(+), 1 deletion(-)

diamond@LAPTOP-O9TI1BHJ MINGW64 /d/abc/exp (main)
$ git log
commit 631d0bf730e671840c63e098f670d24e2f1a677b (HEAD -> main)
Author: diamondful <3207307701@qq.com>
Date:   wed Jun 12 10:46:40 2024 +0800

    提交对触屏鼠标的通用交互操作版本
```

8.UI 的个性化键盘交互控制的设计开发

8.1 分析和设计

经过了上一章对触屏技术设计网页的设计经验后，这一章我们通过更高级的代码来实现对个性化键盘交互功能的通用交互操作初步开发，阐述移动互联时代的用户终端的多样，用例图和 dom 图如下图 8.1。

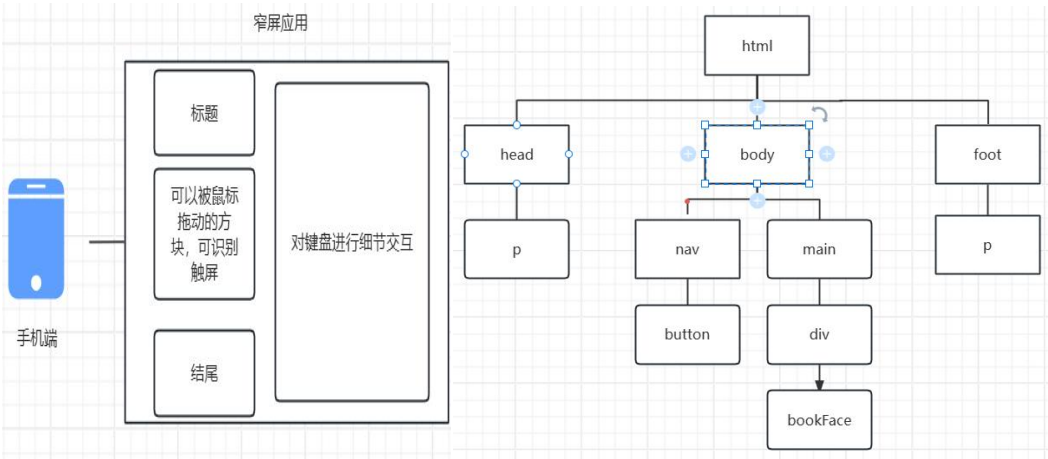


图 8.1

8.2 项目的实现和编程

主要实现功能的代码块如下代码块 8.2.

```
//提出问题：研究利用"keydown"和"keyup"2 个底层事件，实现同时输出按键状态和文本内容
$("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)
});
} //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 $

```

代码块 8.2

8.3 项目的运行和测试

本文此处给出 PC 端用 Chrome 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二维码，运行测试本项目的本次开发的阶段性效果。



图 8.3

8.4 项目的代码提交和版本管理

```
cd /d
Cd /abc
Git add exp/1.6.html
Git commit -m "此版本跟新了对键盘交互的详细设计"
```

```

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   1.6.html

diamond@LAPTOP-O9TI18HJ MINGW64 /d/abc/exp (main)
$ git commit -m"此版本跟新了对键盘交互的详细设计"
[main e89d874] 此版本跟新了对键盘交互的详细设计
1 file changed, 1 insertion(+), 1 deletion(-)

diamond@LAPTOP-O9TI18HJ MINGW64 /d/abc/exp (main)
$ git log
commit e89d8748abf4ce2860bef7234b6a38d1a132f1d0 (HEAD -> main)
Author: diamondful <3207307701@qq.com>
Date:   Wed Jun 12 11:01:02 2024 +0800

    此版本跟新了对键盘交互的详细设计

```

9. 谈谈本项目中的高质量代码

在 1.6.html 中的这段代码让我印象最为深刻如下图代码块 9.1.

```

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 ;
        }
    }
}

```

代码块 9.1

这段代码是一个 JavaScript 函数，根据输入的字符 k 判断该字符是字母、数字还是标点符号。让我们详细讨论一下这段代码的优点：

1. 代码清晰易懂

这段代码使用了适当的变量名和注释，使得代码逻辑清晰，易于理解。通过注释提醒学生需要研究逻辑的作用，有助于促进对代码的深入理解。

2. 使用了临时数组存储标点符号

使用 `puncs` 数组存储了常见的标点符号，这样可以方便地扩展或修改标点符号集合。遍历 `puncs` 数组来检查输入字符是否为标点符号，代码结构清晰。

3. 使用了直观的条件判断

使用了直观的条件判断来检测输入字符是字母、数字还是标点符号，使得代码易于阅读和理解。代码通过逻辑运算符 `&&` 和 `||` 来组合条件判断，减少了代码的复杂度。

4. 考虑了多种情况

函数考虑了输入字符既可以是字母或数字，也可以是标点符号的情况，对不同类型的字符作出了不同的处理。当输入是多个字符时，返回 `false`，这种情况下函数会提前结束并返回结果，避免了不必要的遍历操作。

5. 使用了循环来检查标点符号

使用了 `for...of` 循环遍历标点符号数组，以检查输入字符是否为标点符号，这种方法简洁高效。总体来说，这段代码的优点在于其清晰的逻辑结构，易于理解和扩展。通过对不同类型字符的分类判断，这段代码能够有效地识别输入字符的类型，并返回相应的结果。

10. 用 `gitBash` 工具管理项目代码仓库和 `http` 服务器

10.1 经典 Bash 工具介绍

When we speak of the command line, we are really referring to the shell. The shell is a program that takes keyboard commands and passes them to the operating system to carry out. Almost all Linux distributions supply a shell program from the GNU Project called

bash. The name is an acronym for bourne-again shell, a reference to the fact that bash is an enhanced replacement for sh, the original Unix shell program written by Steve Bourne.^[7]

Like Windows, a Unix-like operating system such as Linux organizes its files in what is called a hierarchical directory structure. This means they are organized in a tree-like pattern of directories (sometimes called folders in other systems), which may contain files and other directories. The first directory in the file system is called the root directory. The root directory contains files and sub directories, which contain more files and sub directories, and so on.^[7]

Bash, 即 **Bourne Again SHell**, 是 **Unix shell** 家族中的一员, 以其强大的功能和灵活性, 成为 **Linux** 系统默认的命令工具。它允许用户在命令行下执行命令, 管理文件和目录, 以及编写脚本, 实现自动化任务。**Bash** 支持通配符和正则表达式, 使得文件名匹配和文本内容搜索变得简单。脚本编写功能让用户能够将一系列命令组合起来, 形成复杂的操作流程, 同时支持条件判断和循环结构, 为流程控制提供了强大的支持。**Bash** 的环境变量和别名功能, 进一步提高了命令行操作的效率。环境变量存储系统和用户配置信息, 而别名简化了常用命令的输入。管道和重定向功能, 允许用户将命令的输出作为另一个命令的输入, 实现数据流的转换, 同时对输入输出流进行重定向, 提供了对标准输入、输出和错误输出的精细控制。**Bash** 的作业控制功能, 使用户能够管理后台运行的作业, 包括启动、暂停、恢复和终止作业, 以及查看当前运行的作业列表。此外, **Bash** 还提供了脚本调试工具, 帮助用户在开发和维护复杂脚本时, 通过设置断点、单步执行和检查变量值, 提高脚本的可维护性和稳定性。**Bash** 的强大之处在于它能够将简单的命令组合成复杂的脚本, 实现自动化和高效的系统管理。它不仅仅是一个命令执行工具, 更是一个功能丰富的编程环境, 使得 **Bash** 成为了命令行工具中的佼佼者, 深受开发者和系统管理员的青睐^[6]。

10.2 通过 gitHub 平台实现本项目的全球域名

要通过 **Git**Hub 搭建自己的域名, 可以按照以下步骤进行操作:

注册域名。首先，在域名注册商（如 GoDaddy、Namecheap 等）注册一个域名。选择一个合适的域名，并完成注册过程。

创建 GitHub 仓库。在 GitHub 上创建一个新的仓库，仓库名称需要符合特定的格式：`yourusername.github.io`，其中 `yourusername` 是你的 GitHub 用户名。

将该仓库设置为公开仓库。添加域名解析，登录到域名注册商的控制面板，找到 DNS 管理或域名设置选项。添加一个新的记录，类型选择为 CNAME，主机或别名为 `www`，将目标或值设置为 `yourusername.github.io`。配置 GitHub Pages，进入你的 GitHub 仓库，在仓库设置 (Repository Settings) 中找到 GitHub Pages 选项。选择使用主分支作为源，并保存设置。等待 DNS 生效 DNS 解析可能需要一些时间来生效，通常需要几分钟甚至几小时。你可以在命令行中使用 `ping yourdomain.com` 来检查域名是否已经指向 GitHub Pages。验证域名，在 GitHub 仓库的根目录下创建一个名为 CNAME 的文件，文件内容为你的域名（例如 `yourdomain.com`）。提交并推送这个文件到你的 GitHub 仓库。启用 HTTPS，在 GitHub 仓库的设置中，找到 Enforce HTTPS 选项，并启用它。这样可以为你的网站提供安全的 HTTPS 连接。

完成以上步骤后，你的域名就会与你的 GitHub Pages 网站关联起来了。现在，当访问你的域名时，将显示你在 GitHub 上托管的网站。

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 *

 diamondful

Repository name *

/

Great repository names are short and memorable. Need inspiration? How about **ideal-funicular** ?

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Create repository

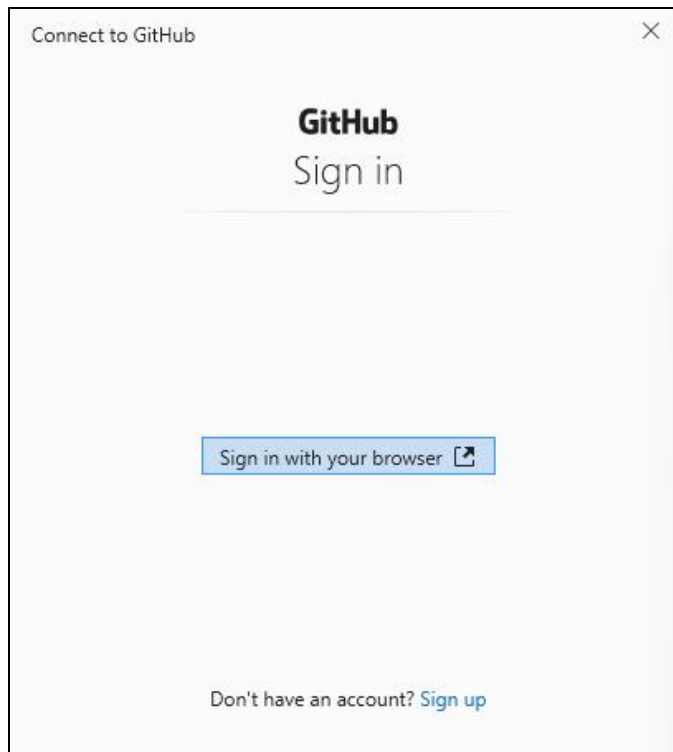
点击窗口右下角的绿色“Create repository”，则可创建一个空的远程代码仓库。

10.4 设置本地仓库和远程代码仓库的链接

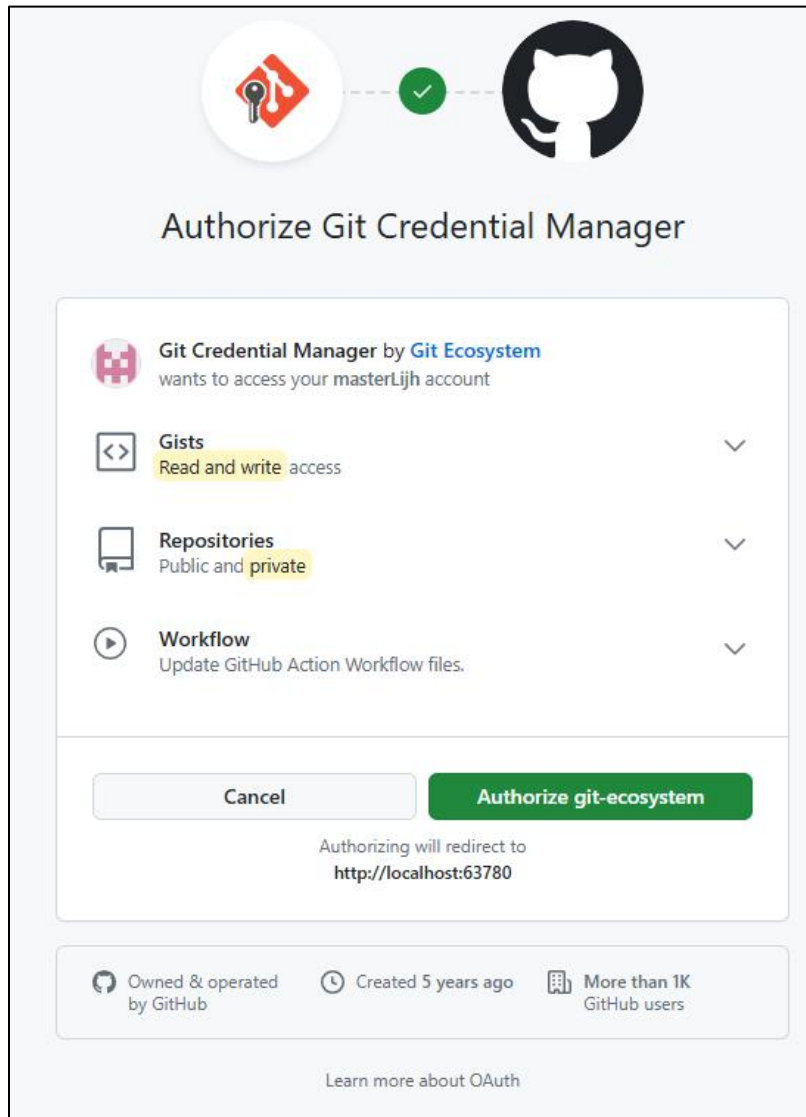
进入本地 webUI 项目的文件夹后，通过下面的命令把本地代码仓库与远程建立密钥链接

```
$ 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/diamondful.github.io.git
$ git push -u origin main
```

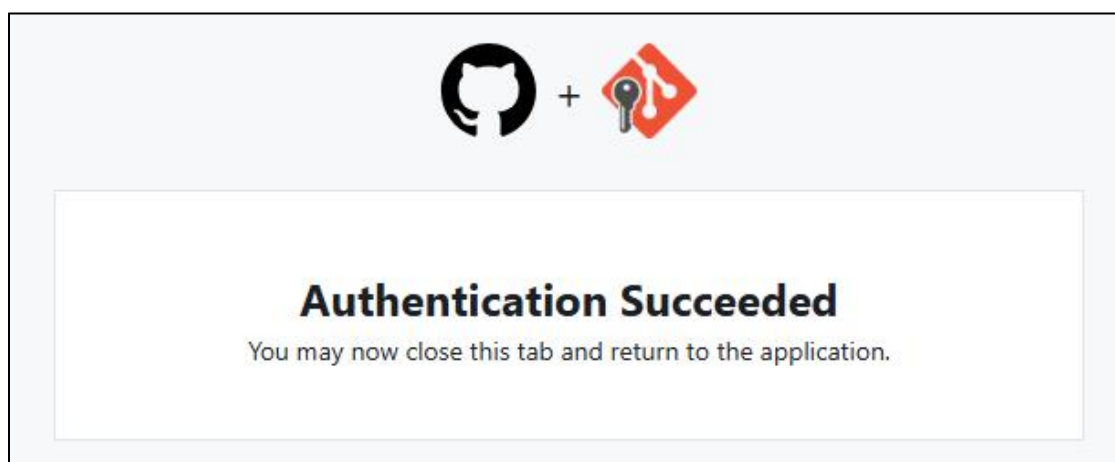
本项目使用 window 平台，gitbash 通过默认浏览器实现密钥生成和记录，第一次链接会要求开发者授权，如下图所示：



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