

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：李登金扬

学    号：20213619

指导教师：李健宏

2024 年 6 月 18 日

## 目录

基于 Web 客户端技术的个性化 UI 的设计和实现 .....	4
(Customized UI design and Programming based on Web client technology)	
.....	4
1. 前言 .....	4
1.1 毕设任务分析 .....	5
1.2 研学计划 .....	5
1.3 研究方法 .....	5
2. 技术总结和文献综述 .....	6
2.1 Web 平台和客户端技术概述 .....	6
2.2.1 历史记录 .....	6
2.2.2 万维网联盟 .....	6
2.2 项目的增量式迭代开发模式 .....	7
3. 内容设计概要 .....	8
3.1 分析和设计 .....	8
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 .....	9
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 项目的运行与测试 .....	12
4.2 项目的代码提交和版本管理 .....	13
5. 应用响应式设计技术开发可适配窄屏和宽屏的 UI .....	14
5.1 项目设计 .....	14
5.3 项目的运行与测试 .....	17
5.3 项目的代码提交和版本管理 .....	18
6. 个性化 UI 设计中对鼠标交互的设计开发 .....	20
6.1 项目设计 .....	20
6.2 项目的运行与测试 .....	21
6.3 项目的代码提交和版本管理 .....	22
7. 对触屏和鼠标的通用交互操作的设计开发 .....	23
7.1 项目设计 .....	23
7.2 项目运行与测试 .....	24
7.3 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 .....	25
8.1 编程与实现 .....	25
8.2 测试与运行 .....	26
8.3 代码提交和版本管理 .....	26
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 .....	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 .....	28
10.4 设置本地仓库和远程代码仓库的链接 .....	29

参考文献: .....	31
-------------	----

# 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 李登金扬

**摘要：**个性化用户界面（UI）在 Web 应用程序中的设计和实现是当前研究的焦点之一。随着互联网的发展和用户需求的多样化，传统的静态和固定 UI 已经不能满足用户的需求。个性化 UI 通过根据用户的偏好、行为和上下文信息来动态调整界面，提升了用户体验和系统的可用性。本文从设计和实现两个方面探讨了基于 Web 客户端技术的个性化 UI 的关键问题。设计方面，个性化 UI 的实现涉及用户数据的收集、分析和利用。通过收集用户的历史操作数据、偏好设置以及实时的行为数据，系统可以建立用户模型，并据此推断用户的兴趣和需求。基于这些信息，系统可以自动调整界面元素的展示顺序、样式和内容，从而优化用户的交互体验。实现方面，个性化 UI 的技术支持主要包括前端页面的动态生成和响应式设计。随着前端技术的不断发展，如 Ajax、JavaScript 框架（如 React、Angular 等）和 CSS 媒体查询，使得在不同终端设备上实现响应式布局变得更加容易和高效。同时，利用服务器端的数据分析和处理，可以实现个性化推荐和动态内容生成，进一步提升用户的满意度和粘性。基于 Web 客户端技术的个性化 UI 设计和实现在提升用户体验、增强系统互动性方面具有重要意义。未来的研究方向包括如何更精准地分析用户行为、如何提高个性化推荐的准确性和实时性，以及如何在保护用户隐私的前提下有效利用用户数据等问题。个性化 UI 的发展将继续推动 Web 应用程序设计和开发的进步，为用户提供更加智能、高效的交互体验。

**关键词：**个性化用户界面；JavaScript 框架；css；web；

## 1. 前言

在当今数字化社会中，Web 应用程序已经成为人们日常生活中不可或缺的一部分。随着互联网技术的迅猛发展和普及，用户对 Web 应用的个性化需求日益增加。个性化用户界面（UI）的设计和实现因其能够根据用户偏好和行为动态调整而备受关注。本文旨在探讨基于 Web 客户端技术的个性化 UI 的设计和实现，以满足用户对个性化体验的需求。在传统的 Web 应用中，用户界面通常是静态和普遍适用的，无法有效地适应不同用户群体的个性化需求。然而，随着信息技术的发展和用户体验设计的深入研究，个性化 UI 成为提升用户满意度和参与度的重要手段之一。个性化 UI 不仅能够根据用户的兴趣、偏好和使用习惯实时调整界面元素，还能提供更加精准和符合个体需求的服务和内容。本文将集中讨论如何利用现代 Web 客户端技术（如 HTML5、CSS3、JavaScript 等）设计和实现个性化 UI。通过分析用户行为数据、偏好信息以及实时反馈机制，我们可以构建一种智能化的用户界面，使其能够根据用户的动态需求和上下文环境进行实时调整和优化。这不仅有助于提升用户体验的个性化程度，还能增强应用程序的用户粘性和

市场竞争力。

## 1.1 毕设任务分析

毕业设计（毕设）任务分析是确保项目成功完成的关键步骤。首先，需要明确项目的背景和目的，包括所属学科领域及研究背景。其次，详细分析任务要求和约束条件，如功能、性能、时间等方面的具体要求和限制。然后，界定项目的范围和限制，确保项目不偏离主题。接着，确定关键任务和阶段性目标，有助于分解复杂任务，更有效地推进项目。最后，制定合理的项目计划和时间安排，评估风险并做好资源规划和管理，以确保项目顺利进行。这些步骤和考虑因素将为毕设的顺利完成提供必要的指导和支持。

## 1.2 研学计划

在计算机科学的毕业设计项目和毕业论文中，制定一个有效的学习计划至关重要。

明确研究领域和课题选择，首先要确定自己感兴趣的研究领域和具体的课题。这可以通过阅读相关文献、参加课程和讨论来达成。学习和研究背景知识，确保掌握和理解与选定课题相关的基础知识和理论。这可能涉及到深入学习特定的编程语言、算法、数据结构、数据库技术等。制定时间表和阶段性目标，制定一个详细的时间表，包括每个阶段的任务和目标。这有助于确保项目在规定时间内按时完成，并有助于分配资源和管理时间。持续学习和反馈机制，毕业设计项目应该是一个持续学习的过程。定期审查进展，并寻求指导教师或导师的反馈和建议，以确保项目在正确的轨道上进行。

## 1.3 研究方法

有效的研究方法是确保毕业设计项目和毕业论文质量的关键。以下是几种常见的研究方法：

（1）实证研究：如果课题适合实证研究，可以设计实验或调查来收集和分析数据，验证假设或解决实际问题。

（2）文献综述：进行系统性的文献综述，总结和分析前人的研究成果和观点，为自己的研究提供理论依据和背景。

（3）案例研究：选择一个典型的案例或实际应用进行深入分析和研究，探讨其在特定环境下的应用效果和影响。

（4）原型开发和评估：如果项目涉及软件开发，可以采用原型开发方法，快速迭代并收集用户反馈，以改进系统的设计和功能。

（5）数学建模和仿真：对于某些技术或理论，可以采用数学建模和仿真方法来验证理论模型的有效性和准确性。

## 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 创立了 Web 的核心技术框架，并建立了 W3C 组织。自 2010 年以来，W3C 推出的 HTML5 国际标准与 ECMA 国际组织维护的 ECMAScript 标准的结合，几乎完美地实现了全球开发者追求统一开发平台的梦想。至今，科学家和 Web 行业依然不懈努力，致力于进一步完善这一伟大而光荣的愿景。在我的毕设项目中，我专注于深入学习和应用 Web 标准及相关技术，以掌握编写跨平台运行的高质量 Web 应用所需的工具和技能。我的目标是打造一套优秀的代码架构，能够在不同平台上无缝运行的应用程序。

#### 2.2.1 历史记录

1989 年，蒂姆·伯纳斯-李爵士发明了万维网（见最初的提案）。他在 1990 年 10 月创造了“万维网”一词，并写下了第一个万维网服务器“httpd”和第一个客户端程序（一个浏览器和编辑器）“世界万维网”。

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，后来成为了 Web 的主要发布格式。他对 uri、HTTP 和 HTML 的最初规范随着网络技术的传播，在更大的圈子中得到了改进和讨论。

#### 2.2.2 万维网联盟

1994 年，在许多公司的敦促下，决定成立万维网联盟。蒂姆·伯纳斯-李爵士开始领导网络联盟团队的基本工作，以培养一个一致的架构，以适应网络标准的快速发展，以构建网站、浏览器和设备，以体验网络所提供的一切。

在创立万维网联盟的过程中，蒂姆·伯纳斯-李爵士创建了一个同行社区。网络技术已经发展得如此之快 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）[3]；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>[6]</sup>。本项目中我一共做了六次项目的开发迭代，如下图 1 所示：

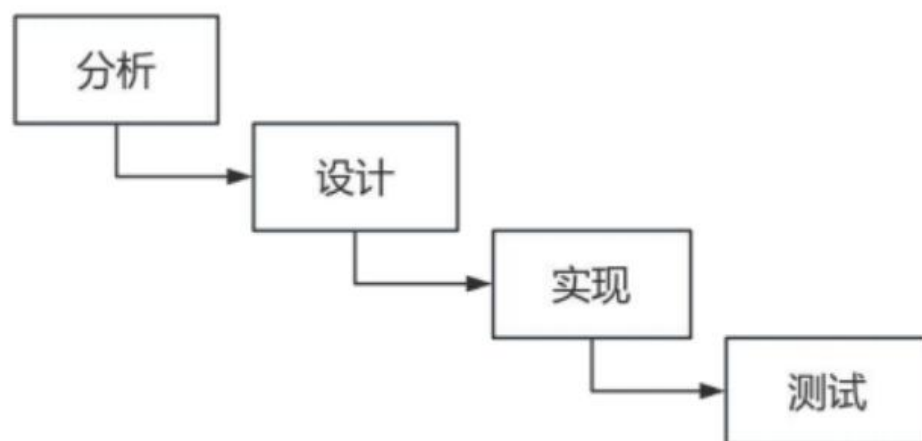


图 1

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

### 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。

#### 3. 项目的实现和编程

(1) HTML 代码编写如下：

```
<header>
    《 深入理解计算机系统（原书第 2 版） 》
</header>
<main>
    我的主题内容
</main>
<footer>
    Copyright 李登金扬 江西科技师范大学 2024-2025
</footer>
```

(2) 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。

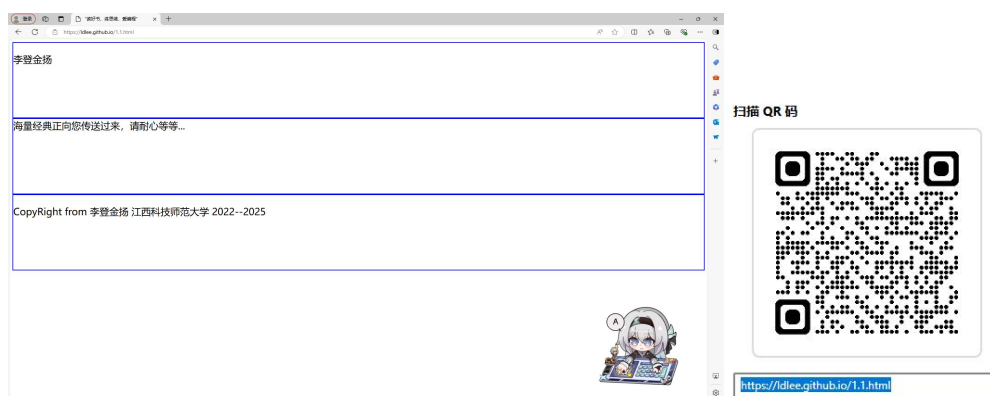


图 4-2 PC 端运行效果图

### 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/abc
$ mkdir webUI
$ cd webUI
$ git init
$ git config user.name 科师大李登金扬

```

```
$ git config user.email 2539686415@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
```

```
it commit -m 项目第一版：“三段论”式的内容设计概要开发
in 7d1828c] 项目第一版：“三段论”式的内容设计概要开发
file changed, 1 insertion(+), 1 deletion(-)
```

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 7d1828c2f4aaa1658c727c784450c5c0bfb249e2 (HEAD -> main)
Author: 科师大李登金扬 <2539686415@qq.com>
Date: Wed Jun 19 13:50:09 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

响应性设计适应显示硬件。在计算机上使用的显示器硬件差别很大，显示器的大小和分辨率取决于成本。设计师并没有选择每个网页的版本，而是选择让网页给出总体布局指南，并允许浏览器选择如何在给定的计算机上显示页面。因此，一个网页并没有提供很多细节。例如，一个网页的作者可以指定一组句子组成一个段落，但作者不能指定细节，如一行的确切长度或是否缩进段落的开头。

允许一个浏览器选择显示细节有一个有趣的结果：当通过两个浏览器或在两个硬件不同的计算机上查看时，一个网页可能会出现不同的外观。如果一个屏幕比另一个宽，一行文本的长度或可以显示的图像的大小就不同。重点是：网页给出了关于所需演示文稿的一般指南；浏览器在显示页面时选择详细信息。因此，当同一网页在两台不同的计算机上显示或通过显示不同时，可能会出现略有不同分析移动互联时代的多样化屏幕的需求。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。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

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

## 4.1 项目的运行与测试

项目的运行截图如图 4-1 所示：

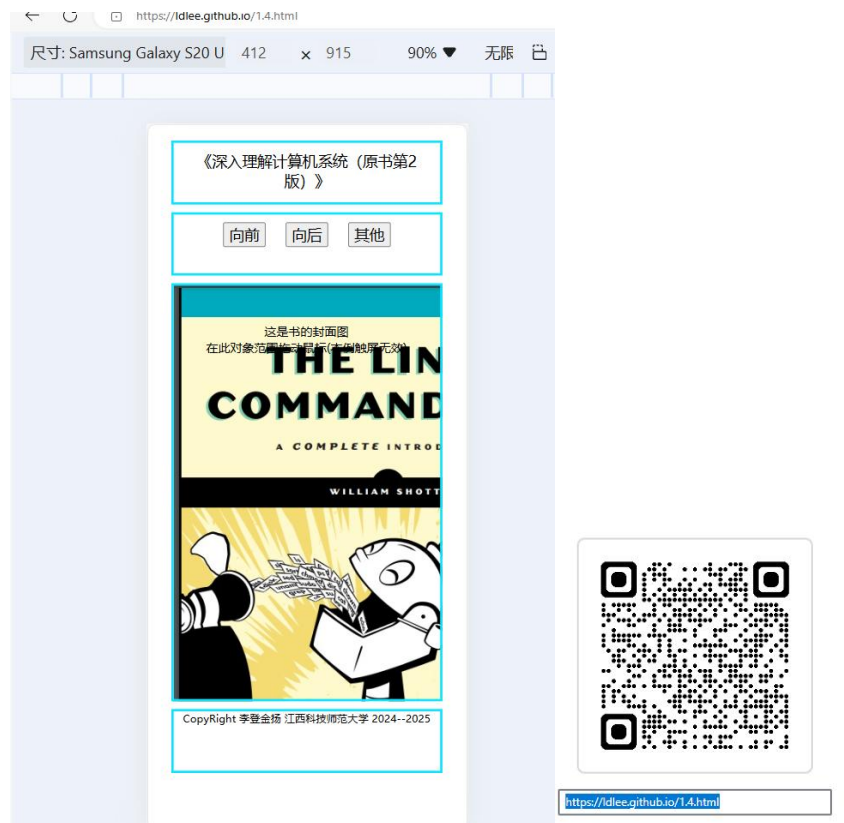


图 4-1 窄屏终端响应

## 4.2 项目的代码提交和版本管理

项目的提交结果如图 4-2 所示：

```
25396@LAPTOP-32BOUK69 MINGW64 /d/abc/webUI (main)
$ git commit -m 项目第二版：实现网页在不同终端的响应式设计，使页面能适应不同尺寸的屏幕
[main a5ffe9f] 项目第二版：实现网页在不同终端的响应式设计，使页面能适应不同尺寸的屏幕
1 file changed, 55 insertions(+), 59 deletions(-)

25396@LAPTOP-32BOUK69 MINGW64 /d/abc/webUI (main)
$ git log
commit a5ffe9fc4848c78970cb6d2a95a27ea29064db1b (HEAD -> main)
Author: 科师大李登金扬 <2539686415@qq.com>
Date: Fri Jun 21 14:43:38 2024 +0800

    项目第二版：实现网页在不同终端的响应式设计，使页面能适应不同尺寸的屏幕
```

4-2 第二版提交

## 5. 应用响应式设计技术开发可适配窄屏和宽屏的 UI

### 5.1 项目设计

在本阶段，我开发出了能够同时适用与宽屏和窄屏设备的 UI 界面，具体的代码块如下：

(1) html 代码如代码块 5-1 所示：

```
</style>
</head>
<body >
  <header>
    <p id="book">
      《深入理解计算机系统（原书第 2 版）》
    </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 李登金扬 江西科技师范大学 2022--2025

  </footer>
  <div id="aid">
    <p>用户键盘响应区</p>
    <p id="keyboard"></p>
  </div>
```

代码块 5-1

(2) CSS 代码如代码块 5-2 所示：

```
<style>
*{
```

```

    margin: 10px;
    text-align: center;
}
header{
    border: 3px solid green;
    height: 10%;
    font-size: 1em;

}
nav{
    border: 3px solid green;
    height: 10%;
}
main{
    border: 3px solid green;
    height: 70%;
    font-size: 0.8em;
    position: relative;
    background-image:url("linuxCMD. jpg");
    background-size:auto;
    background-repeat:no-repeat;
}

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

footer{
    border: 3px solid green;
    height:10%;
    font-size: 0.7em;
}
body{
    position: relative;
}
#aid{
    position: absolute;
    border: 3px solid blue;
    top:0px;
    left:600px;
}
#bookface{

```

```

width: 80%;
height: 80%;
border:1px solid red;
background-color: blanchedalmond;
margin:auto;
}
</style>

```

代码块 5-2

(3) JavaScript 代码如下代码块 5-3 所示:

```

<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+)";
    });
    $("bookface").addEventListener("mouseout",function(ev) {
        //console.log(ev);
        $("bookface").textContent="鼠标已经离开";
    });
    $("body").addEventListener("keypress",function(ev) {
        let k=ev.key;
        let c=ev.keyCode;
        let s1="按键是: ";
        let s2="编码是: "
        $("keyboard").textContent=s1+k+";"+s2+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>

```

代码块 5-3

## 5.3 项目的运行与测试



图 5-1 宽屏用户

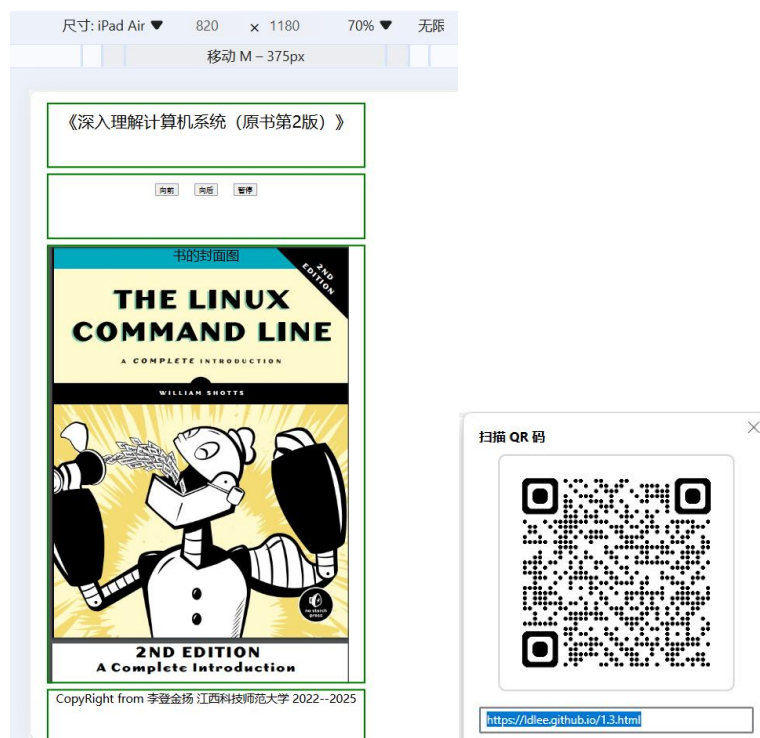


图 5-2 窄屏用户

### 5.3 项目的代码提交和版本管理

项目的提交结果如图 5-3 所示：

```
25396@LAPTOP-32BOUK69 MINGW64 /d/abc/webUI (main)
$ git commit -m 项目第三版：实现页面在窄屏和宽屏设备上的适应性显示
[main ee2add0] 项目第三版：实现页面在窄屏和宽屏设备上的适应性显示
1 file changed, 21 insertions(+), 6 deletions(-)

25396@LAPTOP-32BOUK69 MINGW64 /d/abc/webUI (main)
$ git log
commit ee2add08fae923d146439e52a98198fba7b9d932 (HEAD -> main)
Author: 科师大李登金扬 <2539686415@qq.com>
Date:   Fri Jun 21 15:13:03 2024 +0800

    项目第三版：实现页面在窄屏和宽屏设备上的适应性显示
```

图 5-3 第三版提交

## 6. 个性化 UI 设计中对鼠标交互的设计开发

本阶段深入研究了用户界面的鼠标模型，能显示用户鼠标在界面的具体位置。

### 6.1 项目设计

项目实现的具体代码如下代码块所示：

(1) CSS 代码如代码块 6-1 所示：

```
body{
    position: relative;
}
button{
    font-size:1em;
}
#aid{
    position: absolute;
    border: 3px solid red;
    top:0px;
    left:600px;
}
#bookface{
    position: absolute;
    width: 80%;
    height: 80%;
    left:7% ;
    top: 7% ;
}
```

代码块 6-1

(2) JavaScript 代码如代码块 6-2 所示：

```
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

## 6.2 项目的运行与测试

项目的运行结果如图 6-1 所示：

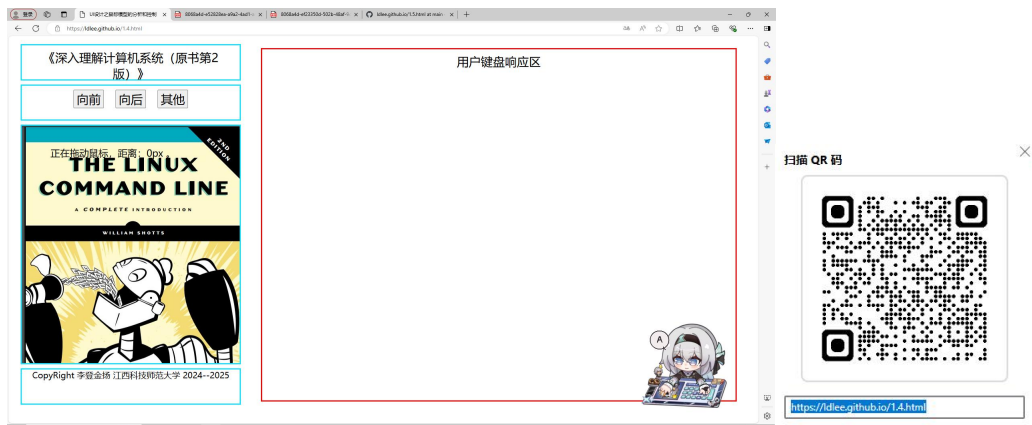


图 6-1 鼠标交互运行结果

## 6.3 项目的代码提交和版本管理

项目的提交结果如下图 6-2 所示：

```
25396@LAPTOP-32BOUK69 MINGW64 /d/abc/WebUI (main)
$ git commit -m 项目第四版：个性化UI设计中鼠标交互的设计开发
[main 77e65d7] 项目第四版：个性化UI设计中鼠标交互的设计开发
1 file changed, 12 insertions(+), 14 deletions(-)

25396@LAPTOP-32BOUK69 MINGW64 /d/abc/WebUI (main)
$ git log
commit 77e65d73a18070613af3a6565a78475823ccb367 (HEAD -> main)
Author: 科师大李登金扬 <2539686415@qq.com>
Date: Fri Jun 21 15:32:18 2024 +0800

    项目第四版：个性化UI设计中鼠标交互的设计开发
```

6-2 第四版提交

## 7. 对触屏和鼠标的通用交互操作的设计开发

本阶段是基于上一个阶段的对鼠标的交互基础，添加了一个用户键盘的相应区。

### 7.1 项目设计

项目的具体实现代码如下所示：

(1) CSS 代码如代码块 7-1 所示：

```
<style>
  body{
    position: relative;
  }
  button{
    font-size:1em;
  }
  #aid{
    position: absolute;
    border: 3px solid blue;
    top:0px;
    left:600px;
  }
  #bookface{
    position: absolute;
    width: 80%;
    height: 80%;
    left:7% ;
    top: 7% ;
  }
</style>
```

代码块 7-1

(2) JavaScript 代码如代码块 7-2 所示：

```
<div id="aid">
  <p>用户键盘响应区</p>
</div>
<script>
  var UI = {};
  if(window.innerWidth>600){
    UI.appWidth=600;
  }else{
```

```

    UI.appWidth = window.innerWidth;
}

$("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

## 7.2 项目运行与测试

项目运行结果如图 7-1 所示：

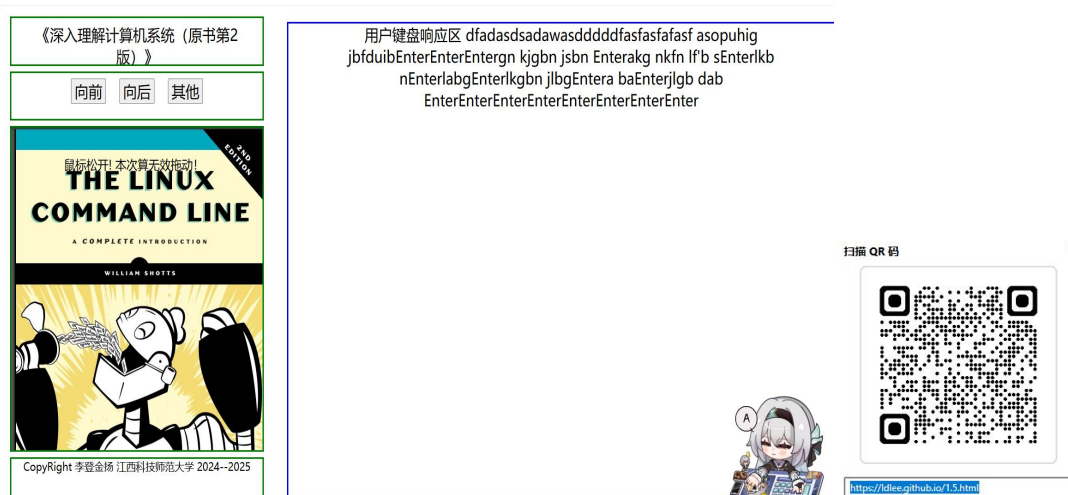


图 7-1 运行结果

## 7.3 代码提交和版本管理

项目的代码提交结果如图 7-2 所示：

```

25396@LAPTOP-32BOUK69 MINGW64 /d/abc/webUI (main)
$ git add 1.5.html

25396@LAPTOP-32BOUK69 MINGW64 /d/abc/webUI (main)
$ git commit -m 第五版：对触屏和鼠标的通用交互操作的设计开发
[main 2fa611f] 第五版：对触屏和鼠标的通用交互操作的设计开发
1 file changed, 2 insertions(+), 2 deletions(-)

25396@LAPTOP-32BOUK69 MINGW64 /d/abc/webUI (main)
$ git log
commit 2fa611f02f9239ce0e98098a4198b427fa1c7083 (HEAD -> main)
Author: 科大李登金扬 <2539686415@qq.com>
Date: Fri Jun 21 15:58:06 2024 +0800

    第五版：对触屏和鼠标的通用交互操作的设计开发

```

图 7-2 第五版提交



## 8. UI 的个性化键盘交互控制的设计开发

当我们探索和利用 `keydown` 和 `keyup` 这两个键盘事件时，我们实际上在利用底层的输入控制能力，为未来的用户界面（UI）功能提供了强大的潜力。这两个事件允许我们捕获用户在键盘上按下和释放键的操作，从而实现各种交互和用户体验的增强。

### 8.1 编程与实现

当我们探索和利用 `keydown` 和 `keyup` 这两个键盘事件时，我们实际上在利用底层的输入控制能力，为未来的用户界面（UI）功能提供了强大的潜力。这两个事件允许我们捕获用户在键盘上按下和释放键的操作，从而实现各种交互和用户体验的增强。为了确保我们的键盘事件能够全面有效地工作，我们通常会将事件监听器设置在 DOM 文档中最大的可视对象——通常是 `<body>` 元素上，如代码块 8-1 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)
});

```

代码块 8-1

## 8.2 测试与运行

测试与运行如图 8-1 所示：



图 8-1 测试

## 8.3 代码提交和版本管理

代码提交结果如图 8-2 所示：

```
$ git add 1.6.html  
25396@LAPTOP-32BOUK69 MINGW64 /d/abc/webUI (main)  
$ git commit -m 项目第六版: UI的个性化键盘交互控制的设计开发  
[main 57cec2e] 项目第六版: UI的个性化键盘交互控制的设计开发  
1 file changed, 2 insertions(+), 2 deletions(-)
```

图 8-2 第六版提交

## 9. 谈谈本项目中的高质量代码

这是一本关于指导计算机的书。如今，电脑和螺丝刀一样常见，但它们相当复杂，让它们做你想让它们做的事情并不总是那么容易。如果你的计算机任务是一个常见的，很容易理解的任务，比如显示你的电子邮件或像一个计算器，你可以打开适当的应用程序，并开始工作。但是对于唯一的或开放式的任务，可能没有应用程序。这就是编程可能会出现的地方。编程是构建一个程序的行为——一组精确的指令，告诉计算机要做什么。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，甚至可以享受到愚蠢的机器能够处理的严格思考，那么编程可能是值得的。它允许你在几秒钟内完成一些永远需要手工完成的事情。这是一种让你的电脑工具做一些它以前不能做的事情的方法。它提供了一个很好的抽象思维的练习。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。

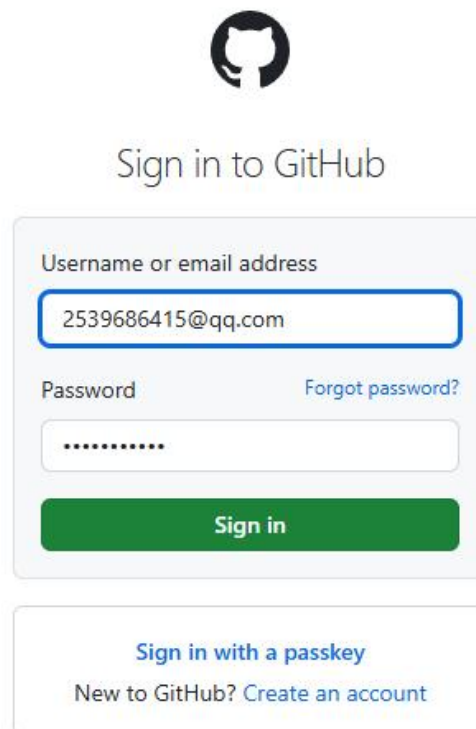
## 10. 用 gitBash 工具管理项目的代码仓库和 http 服务器

### 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·伯恩写的。

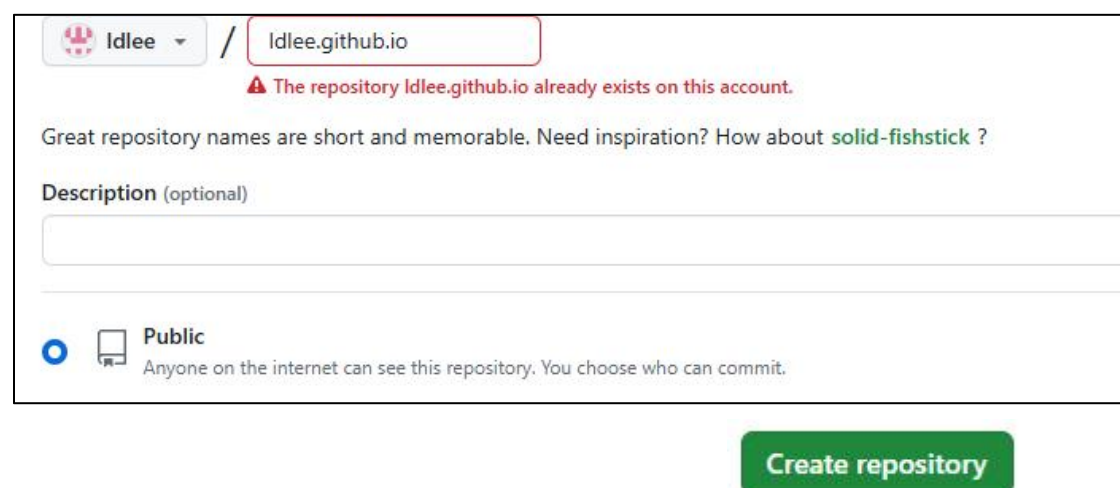
和 Windows 一样，类似 inix 的 Linux 操作系统以分层目录结构组织文件。这意味着它们被组织成一个树状的目录模式（在其他系统中有时称为文件夹），其中可能包含文件和其他目录。文件系统中的第一个目录称为根目录。根目录包含文件和子目录，其中包含更多的文件和子目录，等等。

## 10.2 通过 gitHub 平台实现本项目的全球域名



The image shows the GitHub sign-in page. At the top is the GitHub logo. Below it is the text "Sign in to GitHub". The main form has two input fields: "Username or email address" with the value "2539686415@qq.com" and "Password" with masked characters. There is a "Forgot password?" link next to the password field. A green "Sign in" button is below the fields. At the bottom, there is a link "Sign in with a passkey" and a link "New to GitHub? Create an account".

## 10.3 创建一个空的远程代码仓库



The image shows the GitHub "Create repository" page. At the top, there is a dropdown menu with "Idle" and a search bar with "Idlee.github.io". Below the search bar is a red warning message: "The repository Idlee.github.io already exists on this account." Below the warning is a text input field for the repository name, with a suggestion "solid-fishstick ?". Below the name field is a text input field for the "Description (optional)". Below the description field is a radio button labeled "Public" with a subtext "Anyone on the internet can see this repository. You choose who can commit." At the bottom right is a green "Create repository" button.

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

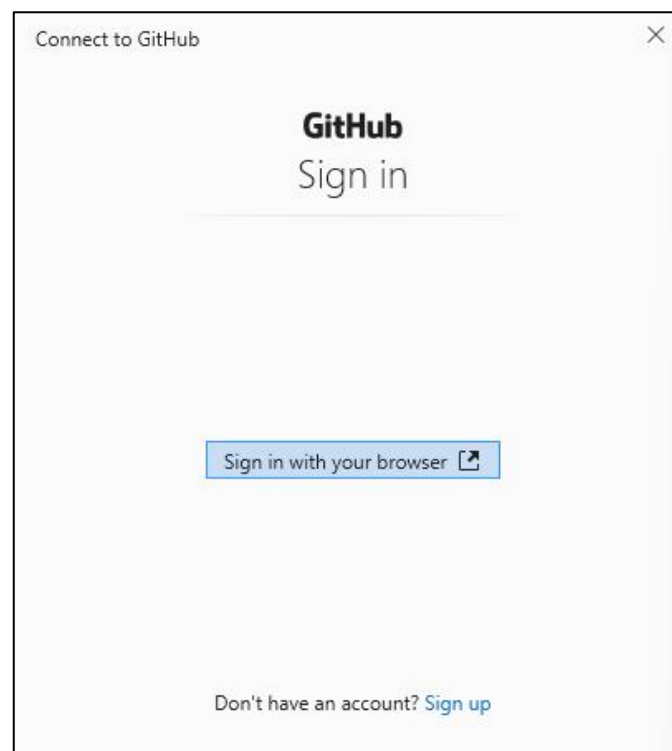
码仓库。

## 10.4 设置本地仓库和远程代码仓库的链接

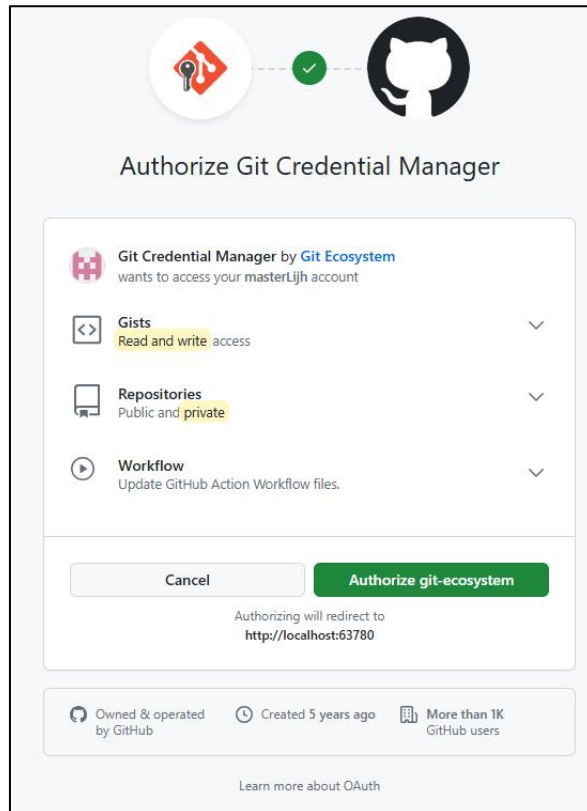
进入本地 webUI 项目的文件夹后,通过下面的命令把本地代码仓库与远程建立密钥链接

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

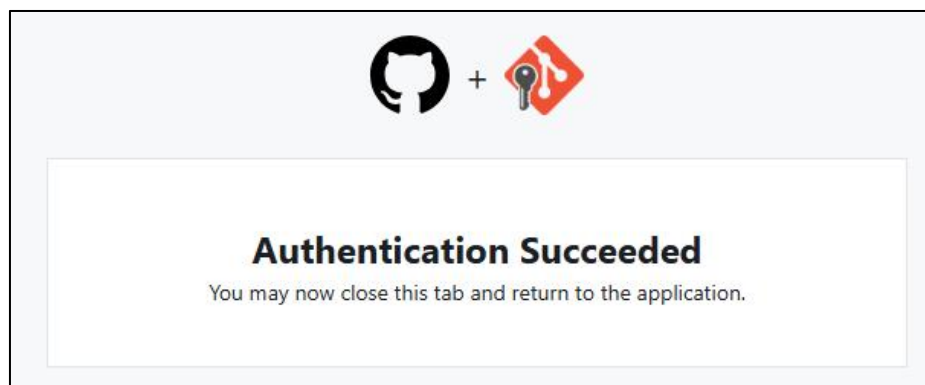
本项目使用 window 平台, gitbash 通过默认浏览器实现密钥生成和记录,第一次链接会要求开发者授权,如下图所示:



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。

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