

江西科技师范大学

课程设计（论文）

题目（中文）：基于 Web 客户端技术的个性化 UI 设计和实现

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

院（系）：元宇宙产业学院

专 业：计算机科学与技术

学生姓名：毛雨欣

学 号：20213571

指导教师：李建宏

2024 年 6 月 18 日

目录

1. 前言	3
1.1 毕设任务分析	3
1.2 研学计划	4
1.3 研究方法	4
2. 技术总结和文献综述	5
2.1 Web 平台和客户端技术概述	5
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	7
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	10
4.1 项目的实现和编程	11
4.2 项目的运行和测试	12
4.3 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	13
5.1 项目的实现和编程	14
5.2 项目的运行和测试	15
5.3 项目的代码提交和版本管理	16
6. 个性化 UI 设计中对鼠标交互的设计开发	17
6.1 项目的实现和编程	17
6.2 项目的运行和测试	19
6.3 项目的代码提交和版本管理	19
7. 对触屏和鼠标的通用交互操作的设计开发	20
7.1 项目的实现和编程	20
7.2 项目的运行和测试	22
7.4 项目的代码提交和版本管理	23
8. UI 的个性化键盘交互控制的设计开发	23
8.1 分析和设计	23
8.2 项目的实现和编程	24
8.3 项目的运行和测试	27
8.4 项目的代码提交和版本管理	28
9. 谈谈本项目中的高质量代码	29
10. 用 gitBash 工具管理项目的代码仓库和 http 服务器	30
10.1 经典 Bash 工具介绍	30
10.2 通过 gitHub 平台实现本项目的全球域名	30
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	31
参考文献:	35
写作指导:	35

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)

摘要：Web 技术以其跨操作系统平台的优势成为了广泛流行的软件开发手段，为了适应移动互联网时代软件项目的前端需求，本项目以 Web 客户端技术为研究学习内容，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: user 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 客户端技术；pc 端；移动端；gitbash

1. 前言

在即将完成学业之际，的确很有必要设计开发一个本专业的作品，来回顾总结本学科专业学习的知识系统，梳理课程体系最核心的东西来体现我们的真实实力。

在该期末设计中，涉及有关的课程理论包括：面向对象的程序设计语言、数据结构的算法、操作系统、软件工程等。在实践层面应用这些核心课程的关键知识，在理解和技术两个维度提升自己的专业性。

在期末设计之前需要多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，最后总结撰写期末论文。在期末项目中，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。

深刻理解计算机系统对专业开发者而言，是非常重要的，对计算机的理解不是浮于表面，更加接近计算机的本质，对任何技术的理解则是能接近技术的底层和基本原理。

1.1 毕设任务分析

毕设任务就是将大学理论的学习在实践层面再做一次综合演练和总结，期间需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，再总结撰写论文。

1.2 研学计划

将该毕设分为两个阶段完成，首先，选择一条适合自己，且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，理解好各个技术之间的关系，在项目中的作用和分工。

第二阶段一般按照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、性能优化。

1.3 研究方法

在撰写论文，和设计项目时，运用到许多的研究方法，如文献法和模型研究法。文献法：利用前人的文字深入学习总结和研究本领域的知识和技术的方法。在我们代码能力成长的任何阶段，我们无法离开文献法，然而承载知识的媒体形式非常丰富包括：书籍、在线文档、期刊等。

模型研究法：这个方法非常的具体。比如我们平时应用到的软硬件，值得我们写代码的角度研究，我们笼统的称为对象，这些对象最终在我们的大脑里就会被理解为抽象的模型，我们再通过分析把这些模型程序化，数据化，最后编写出代码。

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。

2.2.1 历史

1989 年，Sir Tim Berners-Lee 发明了万维网。1990 年 10 月，他创造了“万维网”一词，编写了第一个万维网服务器“httpd”和第一个客户端程序(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语言，成为了 web 的主要发布格式。随着 Web 技术的普及，他最初对 URL、HTTP 和 HTML 的规范在更大范围内得到了改进和讨论。

2.2.2 万维网联盟

1994 年，在许多公司向网络投入越来越多的资源的敦促下，决定成立万维网联盟。Tim Berners-Lee 开始领导网络联盟团队的基本工作，以培养一致的架构，适应构建网站的网络标准的快速进步，浏览器，体验网络所提供的一切的设备。在创建万维网联盟的过程中，Tim Berners-Lee 创建了一个同行社区。Web 技术已经发展得如此之快，因此组件了一个单一的组织来协调 web 标准是至关重要的。Tim 接受了麻省理工学院的邀请来主持 W3C，他有与财团打交道的经验。他从一开始就要求 W3C 具有全球的足迹。

2.2.3 Web 平台和 Web 编程

让我们从 web(万维网的简称)的简要描述开始。大多数人说“Web”而不是“WorldWideWeb”，我们将遵循这个惯例。Web 是被称为网页的文档集合，它们(大部分)被世界各地的计算机用户共享。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。“内容”指的是文本、图片和用户输入机制，如文本框和按钮。

Web 编程是一个很大的领域，不同类型的 web 编程使用不同的工具来实现。所有的工具都使用核心语言 HTML，所以几乎所有的 web 编程书籍都在一定程度上描述了 HTML。这本教科书深入地覆盖了 HTML5，CSS 和 JavaScript。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，所有网页技术都在最终用户的计算机(客户端计算机)上执行。

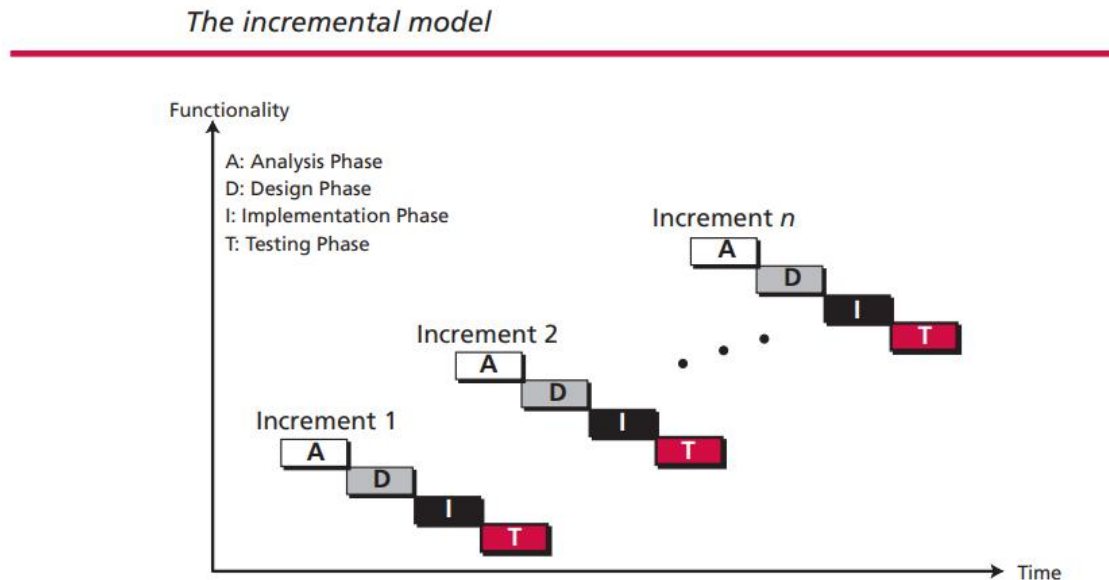
Web 应用的程序设计体系由三大语言有机组成：HTML, CSS, JavaScript。这三大语言的组合也体现了人类社会化大生产分工的智慧，可以看作用三套相对独立体系实现了对一个信息系统的描述和控制，可以总结为：HTML 用来描述结构（Structure）、CSS 用来描述外表（presentation）、Javascript 用来描述行为（Behavior）[3]；这也可以用经典的 MVC 设计模式来理解 Web 平台架构的三大基石，Model 可以理解为 HTML 标记语言建模，View 可以理解为用 CSS 语言来实现外观，Controller 则可理解为用 JavaScript 结合前面二个层次，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[5]。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：



增量模型

在增量模型中，软件的开发分一系列步骤进行。开发人员首先完成整个系统的简化版本。此版本代表整个系统，但不包括详细信息。图显示了增量模型概念。在第二个版本中，增加了更多的细节，但有些还没有完成，并再次测试系统。如果有问题，开发人员知道问题出在新的功能上。在现有系统正常工作之前，它们不会添加更多功能。此过程将继续，直到添加了所有必需的功能。

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。如图 4-1 用例图所示：

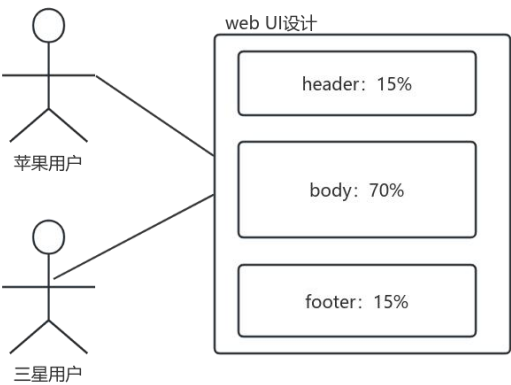


图 4-1 用例图

3.2 项目的实现和编程

一、HTML 代码编写如下：

```

<header>
  <p id="book">
    现代电影赏析
  </p>
</header>
<nav>
  <button>向前</button>
  <button>暂停</button>
  <button>向后</button>
</nav>
<main id = 'main'>
  "赏佳影、品内涵、写影评"
</main>
<footer>
  <p id="statusInfo">
    Copyright from 毛雨欣 江西科技师范大学 2022--2025
  </p>
</footer>

```

二、CSS 代码编写如下：

```

<style>
  *{
    margin: 10px;
    text-align: center;
  }
  header{
    border: 2px solid #4CAF50;
    height: 80px;
  }
  main{
    border: 2px solid #4CAF50;
    height: 300px;
  }
  nav{
    border: 2px solid #4CAF50;
    height: 50px;
  }
  footer{
    border: 2px solid #4CAF50;
    height: 50px;
  }
</style>

```


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



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 江科师@毛雨欣
$ git config user.email 3388564824@qq.com
$ touch 1-1.html myCss.css
```

编写好 index.html 的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add 1-1.html
$ 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：

```
86187@LAPTOP-2TRRBLJE MINGW64 /d/maoyx/GLJ/abc/example (master)
$ git add 1-1.html

86187@LAPTOP-2TRRBLJE MINGW64 /d/maoyx/GLJ/abc/example (master)
$ git commit -m "三段论式的内容设计概要开发"
[master ee28e2d] 三段论式的内容设计概要开发
1 file changed, 57 insertions(+)
create mode 100644 1-1.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 ee28e2d128682e4c1529c5f0a09500c5636a3b6b
Author: BLUE12356 <3388564824@qq.com>
Date: Tue Jun 18 17:26:28 2024 +0800

    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

响应式设计-容纳显示硬件

与计算机一起使用的显示硬件变化很大，显示器的尺寸和分辨率取决于成本。而不是针对每种类型的显示器具有每个网页的版本，设计人员选择使网页提供一般布局准则，并允许浏览器选择如何在给定计算机显示页面。因此，网页不会给出许多细节。例如，网页的作者可以指定一组句子形成一个段落，但是作者不能给定详细信息，例如行的准确长度或是否缩进段落的开头。允许浏览器选择显示详细信息会产生一个有趣的结果：当通过两个浏览器或在具有不同硬件的两台计算机上查看网页时，网页的外观可能会有所不同。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：网页提供了有关所需演示文稿的一般准则；浏览器在显示页面时选择详细信息。因此，当在两台不同的计算机上或通过不同的浏览器显示相同的网页时，其外观可以略有不同。

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。

4.1 项目的实现和编程

一、HTML 代码编写如下：

```
<header>
  <p id="book">
    | 现代电影赏析
  </p>
</header>
<nav>
  <button>向前</button>
  <button>暂停</button>
  <button>向后</button>
</nav>
<main id = 'main'>
  | "赏佳影、品内涵、写影评"
</main>

<footer>
  <p id="statusInfo">
    | 毛雨欣 江西科技师范大学 2025
  </p>
</footer>
```

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。

二、CSS 代码编写如下：

```

<style>
*{
margin: 10px;
text-align: center;
}
header{
border: 2px solid #hsl(111, 76%, 64%);
height: 15%;
font-size: 1.66em;
}
main{
border: 2px solid #hsl(111, 76%, 64%);
height: 70%;
font-size: 1.2em;
background-image: url(../lesson/CS.jpg);
background-size: cover;
}
nav{
border: 2px solid #hsl(111, 76%, 64%);
height: 10%;
font-size: 1.1em;
}
footer{
border: 2px solid #hsl(111, 76%, 64%);
height: 5%;
}

```

代码块 4-2

4.2 项目的运行和测试



图 4-3 pc 端运行效果图



图 4-3 移动端二维码

4.3 项目的代码提交和版本管理

编写好 1-2.html 的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add 1-2.html  
$ git commit -m 项目第二版：窄屏终端的响应式设计
```

```
86187@LAPTOP-2TRRBLJE MINGW64 /d/maoyx/GLJ/abc/example (master)  
$ git add 1-2.html  
  
86187@LAPTOP-2TRRBLJE MINGW64 /d/maoyx/GLJ/abc/example (master)  
$ git commit -m "窄屏终端的响应式设计"  
[master 300c604] 窄屏终端的响应式设计  
1 file changed, 69 insertions(+)  
create mode 100644 1-2.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 300c604fcc62a2a81ea9c844bae7f0b372b9862e  
Author: BLUE12356 <3388564824@qq.com>  
Date: Tue Jun 18 17:28:13 2024 +0800  
  
窄屏终端的响应式设计
```

5. 应用响应式设计技术开发可适配窄屏和宽屏的 UI

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，本阶段开发了可适配的窄屏和宽屏，如果窗口的宽度低于 1000，将不会出现“aid”，如果窗口的宽度高于 1000，“aid”的宽度等于窗口的宽度减去 UI 的宽度再减去 30px。尝试对鼠标设计 UI 控制，按下鼠标键，坐标将随鼠标的位置随之改变。

5.1 项目的实现和编程

一、HTML 代码编写如下：

```
<header>
  <p id="book">
    | 现代电影赏析
  </p>
</header>
<nav>
  <button>向前</button>
  <button>暂停</button>
  <button>向后</button>
</nav>
<main id = 'main'>
  <div id="bookface">
    | "赏佳影、品内涵、写影评"
  </div>
</main>
<footer>
  <p id="statusInfo">
    | 毛雨欣 江西科技师范大学 2025
  </p>
</footer>
```

二、CSS 代码编写如下：


```
<style>
*{
  margin: 10px;
  text-align: center;
}
header{
  border: 2px solid hsl(111, 76%, 64%);
  height: 15%;
  font-size: 1.66em;

}
main{
  border: 2px solid hsl(111, 76%, 64%);
  height: 70%;
  font-size: 1.2em;
  background-image: url(../lesson/CS.jpg);
  background-size: cover;
}
nav{
  border: 2px solid hsl(111, 76%, 64%);
  height: 10%;
  font-size: 1.1em;
}
footer{
  border: 2px solid hsl(111, 76%, 64%);
  height: 5%;
}
#aid{
  position: absolute;
  border: 3px solid pink;
  top:0px;
  left:600px;
}
#bookface{
  width: 80%;
  height:80%;
  border: 1px solid yellow;
  background-color: antiquewhite;
  margin: auto;
}
</style>
```

5.2 项目的运行和测试



图 5-3. pc 端运行效果图



图 5-3.移动端二维码

5.3 项目的代码提交和版本管理

编写好 1-3.html 的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add 1-3.html
$ git commit -m 项目第三版：应用响应式设计技术开发可适配宽屏和窄屏的 UI
```

```
86187@LAPTOP-2TRRBLJE MINGW64 /d/maoyx/GLJ/abc/example (master)
$ git add 1-3.html

86187@LAPTOP-2TRRBLJE MINGW64 /d/maoyx/GLJ/abc/example (master)
$ git commit -m "应用响应式设计技术开发可适配宽屏和窄屏的UI"
[master (root-commit) 3871286] 应用响应式设计技术开发可适配宽屏和窄屏的
UI
1 file changed, 138 insertions(+)
create mode 100644 1-3.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 387128670748c4cc9c5ebd3c143a579fb27fa686
Author: BLUE12356 <3388564824@qq.com>
Date: Tue Jun 18 17:18:53 2024 +0800
```

```
应用响应式设计技术开发可适配宽屏和窄屏的UI
:
```

6. 个性化 UI 设计中对鼠标交互的设计开发

用汉语言来描述我们是如何实现的，与上一阶段比较，该阶段通过改变 **body** 对象的字体大小，这个属性可以影响其后代。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。在对鼠标进行控制时，鼠标按下，则为有效拖动，坐标发生变化，鼠标松开，则无效拖动，鼠标停止变化。

6.1 项目的实现和编程

一、HTML 代码编写如下：

```
<header>
| <p id="book">
|   现代电影赏析
| </p>
</header>
<nav>
| <button>向前</button>
| <button>暂停</button>
| <button>向后</button>
</nav>
<main id = 'main'>
| <div id="bookface">
|   "赏佳影、品内涵、写影评"
| </div>
</main>
<footer>
| <p id="statusInfo">
|   毛雨欣 江西科技师范大学 2025
| </p>
</footer>
```

二、CSS 代码编写如下：

```
<style>
*{
margin: 10px;
text-align: center;
```

```

}
header{
  border: 2px solid hsl(111, 76%, 64%);
  height: 15%;
  font-size: 1.66em;

}
main{
  border: 2px solid hsl(111, 76%, 64%);
  height: 70%;
  font-size: 1.2em;
  background-image: url(../lesson/CS.jpg);
  background-size: cover;
}
nav{
  border: 2px solid hsl(111, 76%, 64%);
  height: 10%;
  font-size: 1.1em;
}
footer{
  border: 2px solid hsl(111, 76%, 64%);
  height: 5%;
}
#aid{
  position: absolute;
  border: 3px solid pink;
  top:0px;
  left:600px;
}
#bookface{
  width: 80%;
  height:80%;
  border: 1px solid yellow;
  background-color: antiquewhite;
  left:7% ;
  top: 7% ;
}
</style>

```

6.2 项目的运行和测试

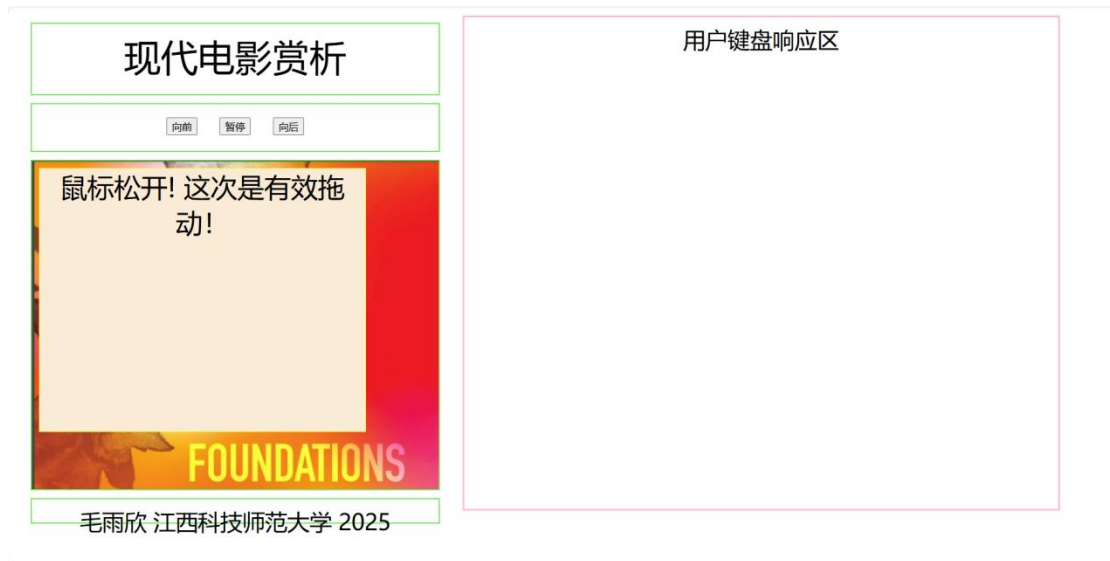


图 6-3. pc 端运行效果图



图 6-3.移动端二维码

6.3 项目的代码提交和版本管理

编写好 1-4.html 的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add 1-4.html
```

```
$ git commit -m 项目第四版：个性化 UI 设计中鼠标交互的设计开发
```

```
86187@LAPTOP-2TRRBLJE MINGW64 /d/maoyx/GLJ/abc/example (master)
$ git add 1-4.html

86187@LAPTOP-2TRRBLJE MINGW64 /d/maoyx/GLJ/abc/example (master)
$ git commit -m "个性化UI设计中对鼠标交互的设计开发"
[master fcec33f] 个性化UI设计中对鼠标交互的设计开发
1 file changed, 170 insertions(+)
create mode 100644 1-4.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 fcec33f9e4996f341ab95222c7f45c42b2b03d1e
Author: BLUE12356 <3388564824@qq.com>
Date: Tue Jun 18 17:31:32 2024 +0800

    个性化UI设计中对鼠标交互的设计开发
```

7. 对触屏和鼠标的通用交互操作的设计开发

用汉语言来描述我们是如何实现的，与上一阶段比较，该阶段多添加了一个鼠标响应区域，在一定范围内拖动鼠标/滑动触屏 拖动/滑动超过 100 像素，视为有效 UI 互动！

7.1 项目的实现和编程

一、HTML 代码编写如下：

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

  <main id='main'>
    <div id="bookface">
      | "赏佳影、品内涵、写影评" <br>
      | 在此对象范围拖动鼠标/滑动触屏<br>
      | | 拖动/滑动超过100像素，视为有效UI互动!
    </div>
  </main>

</footer>

  毛雨欣 江西科技师范大学 2025

</footer>
```

二、 CSS 代码编写如下：

```
<style>
*{
    margin: 10px;
    text-align: center;
}
header{
    border: 3px solid hsl(111, 76%, 64%);
    height: 10%;
    font-size: 1em;
}
nav{
    border: 3px solid hsl(111, 76%, 64%);
    height: 10%;
}
main{
    border: 3px solid hsl(111, 76%, 64%);
    height: 70%;
    font-size: 0.8em;
    background-image: url(../lesson/CS.jpg);
    position: relative;
}

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

footer{
    border: 3px solid hsl(111, 76%, 64%);
    height: 10%;
    font-size: 0.7em;
}
body{
    position: relative;
}
button{
    font-size: 1em;
}
#aid{
    position: absolute;
    border: 3px solid pink;
}
```

```

top:0px;
left:600px;
}
#bookface{
position: absolute;
width: 80%;
height: 80%;
border: 1px solid yellow;
background-color: antiquewhite;
left:7% ;
top: 7% ;
}
</style>

```

7.2 项目的运行和测试



图 7-3. pc 端运行效果图



图 7-3.移动端二维码

7.4 项目的代码提交和版本管理

编写好 1-5.html 的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add 1-5.html
$ git commit -m 项目第五版：对触屏和鼠标的通用交互操作的设计开发
```

```
86187@LAPTOP-2TRRBLJE MINGW64 /d/maoyx/GLJ/abc/example (master)
$ git add 1-5.html

86187@LAPTOP-2TRRBLJE MINGW64 /d/maoyx/GLJ/abc/example (master)
$ git commit -m "对触屏和鼠标的通用交互操作的设计开发"
[master 6e01a19] 对触屏和鼠标的通用交互操作的设计开发
1 file changed, 214 insertions(+)
create mode 100644 1-5.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 6e01a1936fccf27b3944a4d363690d689ea24734
Author: BLUE12356 <3388564824@qq.com>
Date: Tue Jun 18 17:32:33 2024 +0800

    对触屏和鼠标的通用交互操作的设计开发
```

8. UI 的个性化键盘交互控制的设计开发

8.1 分析和设计

这一步是项目的最终开发，本项目在该阶段设计了 UI 宽度，如果 UI 宽度大于 600，则 UI 宽度只显示 600。UI 高度等于屏幕高度。并且在该阶段，尝试了对鼠标和触屏同时进行设计 UI 控制。拖动鼠标，坐标发生变化，松下鼠标，则视为拖动无效。

如图 8-1 用例图所示：

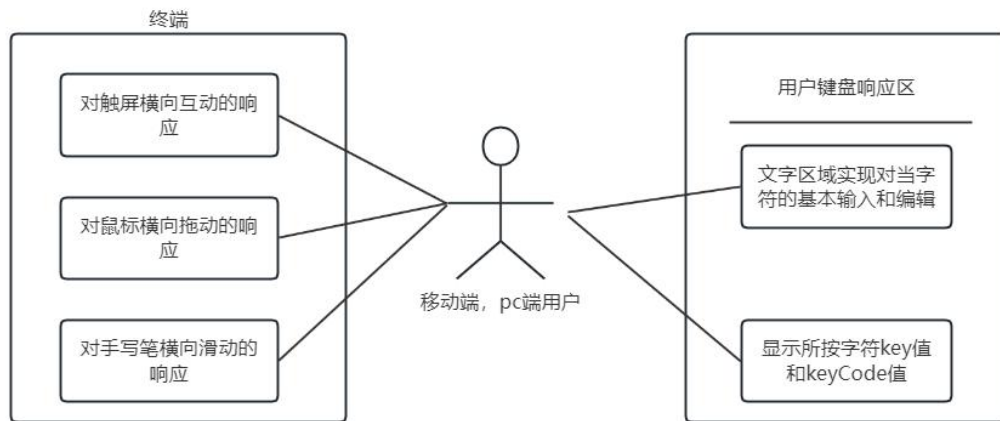


图 8-1 用例图

8.2 项目的实现和编程

一、HTML 代码编程如下：

```

<header>
|   现代电影赏析
</header>
<nav>
|   <button>向前</button>
|   <button>暂停</button>
|   <button>向后</button>
</nav>
<main id="main">
|   <div id="bookface">
|       "赏佳影、品内涵、写影评"
|   </div>
</main>

<footer>
|   毛雨欣 江西科技师范大学 2022--2025
</footer>

```

二、CSS 代码编程如下：


```

<style>
* {
    margin: 10px;
    text-align: center;
}

body {
    position: relative;
}

header {
    height: 15%;
    border: 2px solid #hsl(111, 76%, 64%);
    font-size: 1.6em;
}

main {
    height: 70%;
    border: 2px solid #hsl(111, 76%, 64%);
    font-size: 1.2em;
    background-image: url(../lesson/CS.jpg);
    background-size: cover;
    position: relative;
}

#bookface {
    width: 80%;
    height: 80%;
    border: 1px solid #yellow;
    background-color: #antiquewhite;
    position: absolute;
    left: 8%;
    top: 8%;
}

nav {
    border: 2px solid #hsl(111, 76%, 64%);
    height: 10%;
    font-size: 1.1em;
}

footer {
    min-height: 5%;
    border: 2px solid #hsl(111, 76%, 64%);
}

```

```

#aid {
    position: absolute;
    left: 600px;
    top: 0;
    border: 3px solid pink;
}

#outputText {
    color: hsl(111, 76%, 64%);
    word-break: break-all;
    border: 1px solid pink;
    height: 10%;
    width: 95%;
}

#keyStatus {
    position: absolute;
    bottom: 0;
    border: 1px solid pink;
    width: 90%;
    height: 10%;
}
</style>

```

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("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";
    $("#outputText").textContent += key;
});

```

```

function $(element) {
    if (typeof element !== 'string'
        || element === ''
        || element === null
        || element === undefined) {
        throw new Error('参数错误');
    }

    let dom = document.getElementById(element);
    if (!dom) {
        // 如果不存在
        dom = document.querySelector(element);
    }
    if (dom) {
        return dom;
    }

    // 如果不存在，就抛异常
    throw ('id或选择器不存在');
}

function setBodyStyle(baseFont, appHeight, appWidth) {
    // 通过修改body对象的字体大小，可以遗传给子代，从而实现自己的响应式设计
    document.body.style.fontSize = baseFont + 'px';
    // 把body的高度设置为屏幕的高度，实现了纵向全屏
    // 通过CSS对body子对象的高度百分比进行分配，从而达到响应式设计的目标。
    document.body.style.height = appHeight - 70 + 'px';
    document.body.style.width = appWidth + 'px';
}

/**
 * 给aid设置样式
 * @param { number } appHeight 设备高度
 * @param { number } appWidth 设备宽度
 */
function setAidStyle(appWidth, appHeight) {
    let aid = $('aid');
    if (window.innerWidth <= 1000) {
        aid.style.display = 'none';
    }
    aid.style.width = window.innerWidth - appWidth - 30 + 'px';
    aid.style.height = appHeight - 62 + 'px';
}

```

8.3 项目的运行和测试

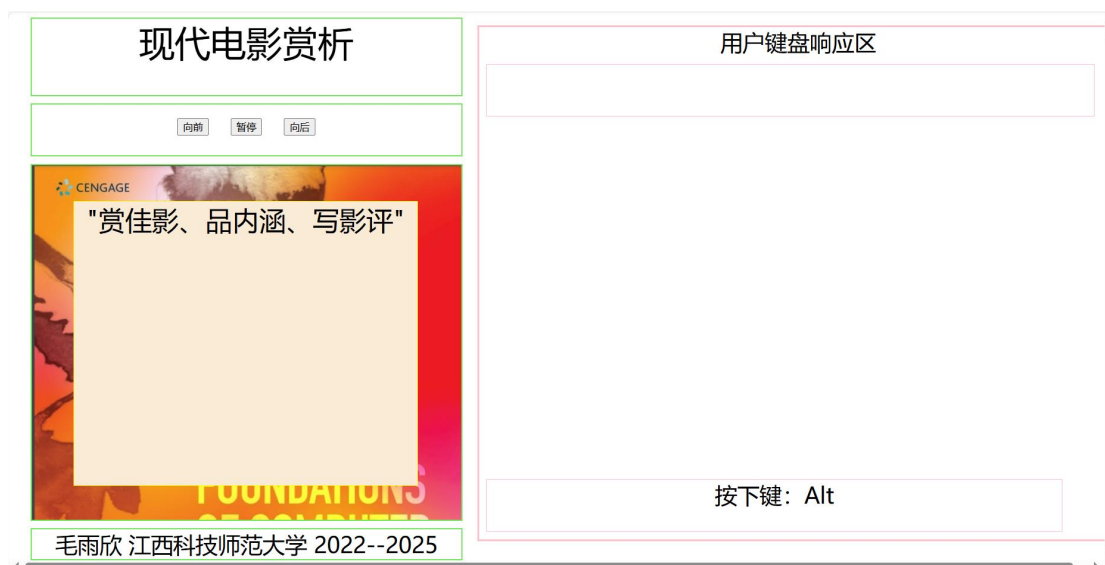


图 8-3. pc 端运行效果图



图 8-3. 移动端二维码

8.4 项目的代码提交和版本管理

编写好 1-6.html 的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add 1-6.html
```

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

```
86187@LAPTOP-2TRRBLJE MINGW64 /d/maoyx/GLJ/abc/example (master)
$ git add 1-6.html

86187@LAPTOP-2TRRBLJE MINGW64 /d/maoyx/GLJ/abc/example (master)
$ git commit -m "UI的个性化键盘交互控制的设计开发"
[master a8c78e0] UI的个性化键盘交互控制的设计开发
1 file changed, 268 insertions(+)
create mode 100644 1-6.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 a8c78e05d01623b992558a2a45fea2d7cc887d34
Author: BLUE12356 <3388564824@qq.com>
Date: Tue Jun 18 17:33:30 2024 +0800

    UI的个性化键盘交互控制的设计开发
```

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。

尝试对鼠标设计 UI 控制，代码编写如下：

```
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+")";
});

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 ;
        }
    }
}
```

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

当我们提到命令行时，我们实际上指的是 shell。Shell 是一个接受键盘命令并将它们传递给操作系统执行的程序。几乎所有 Linux 发行版都提供来自 GNUProject 的 shell 程序 bash。这个名字是 Bourne-Againshell 的首字母缩写，它是 bash 对 sh 的增强替换，sh 是由 SteveBourne 编写的最初的 Unixshell 程序。

与 windows 类似，类似 Unix 的操作系统(如 Linux)以所谓的分层目录结构来组织其文件。这意味着它们以树状目录模式(在其他系统中有时称为文件夹)进行组织，其中可能包含文件的其他目录。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


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 *

 masterLijh

 /

Repository name *

✓ userName.github.io is available.

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

Create repository

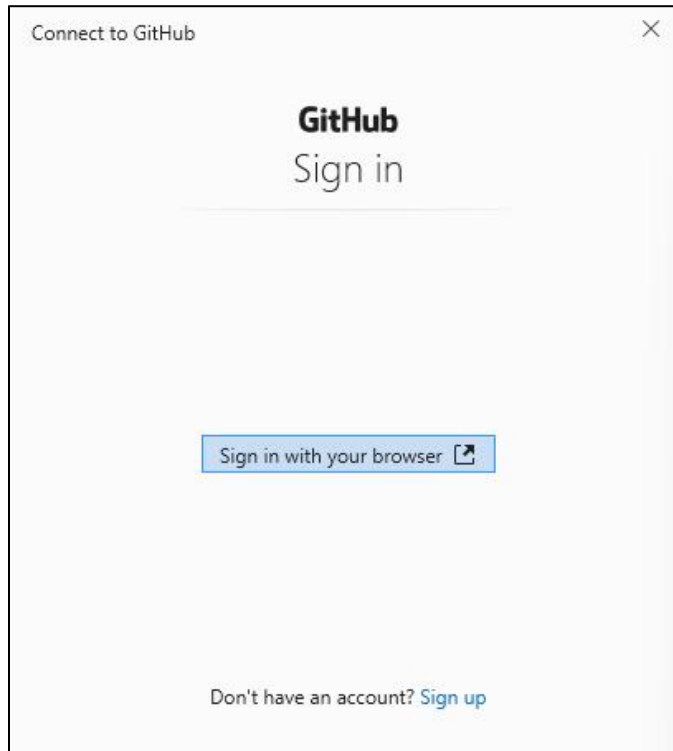
点击窗口右下角的绿色“Create repository”，则可创建一个空的远程代码仓库。

10.4 设置本地仓库和远程代码仓库的链接

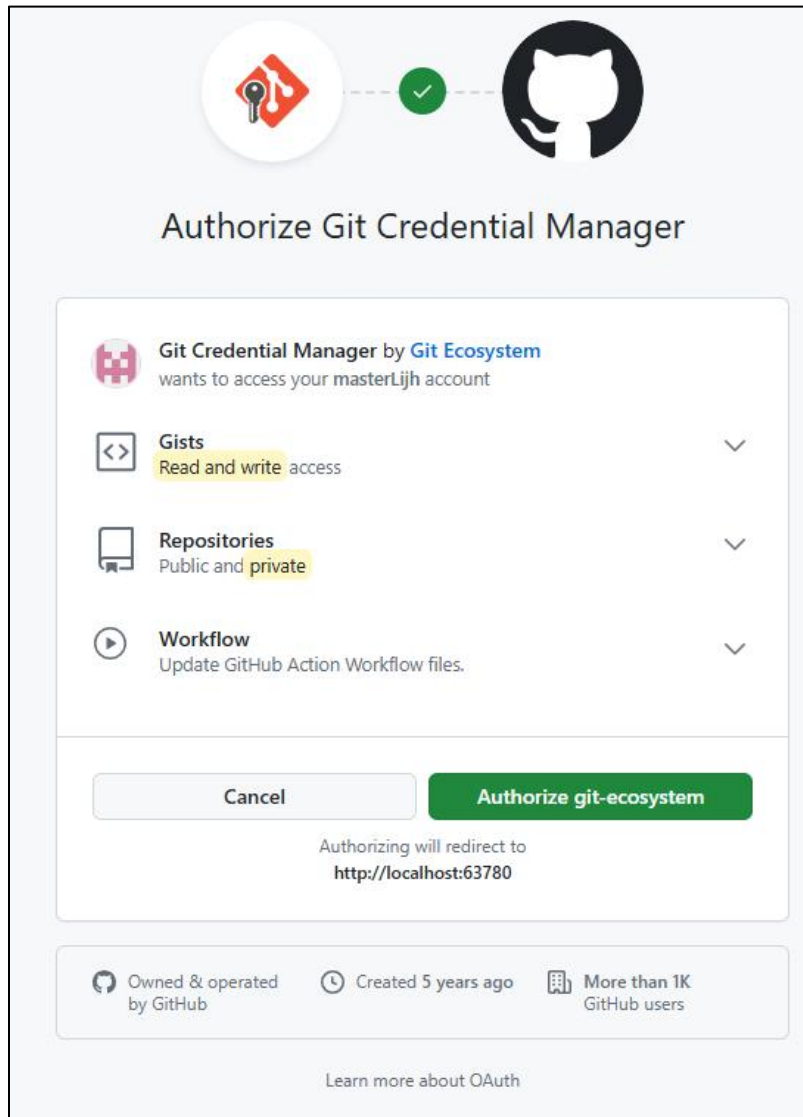
进入本地 webUI 项目的文件夹后，通过下面的命令把本地代码仓库与远程建立密钥链接

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

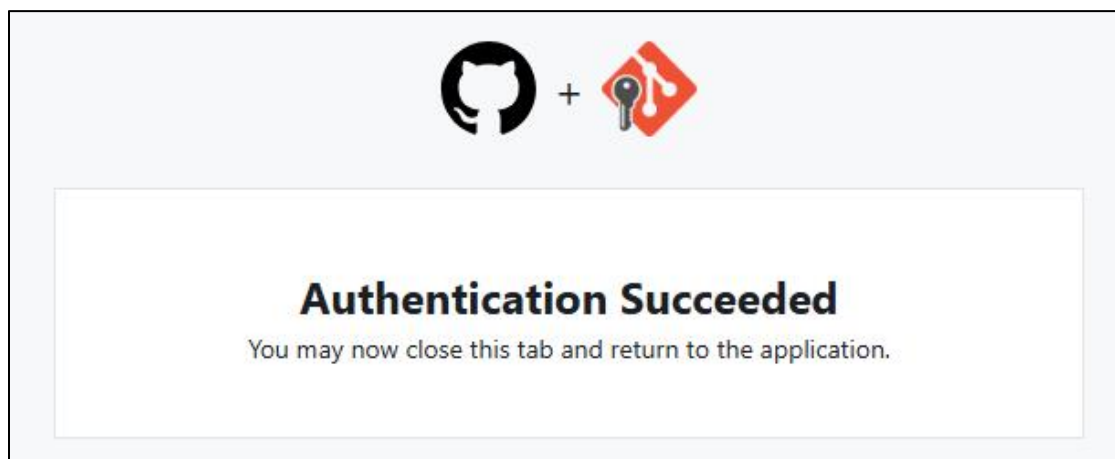
本项目使用 window 平台，gitbash 通过默认浏览器实现密钥生成和记录，第一次链接会要求开发者授权，如下图所示：



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：

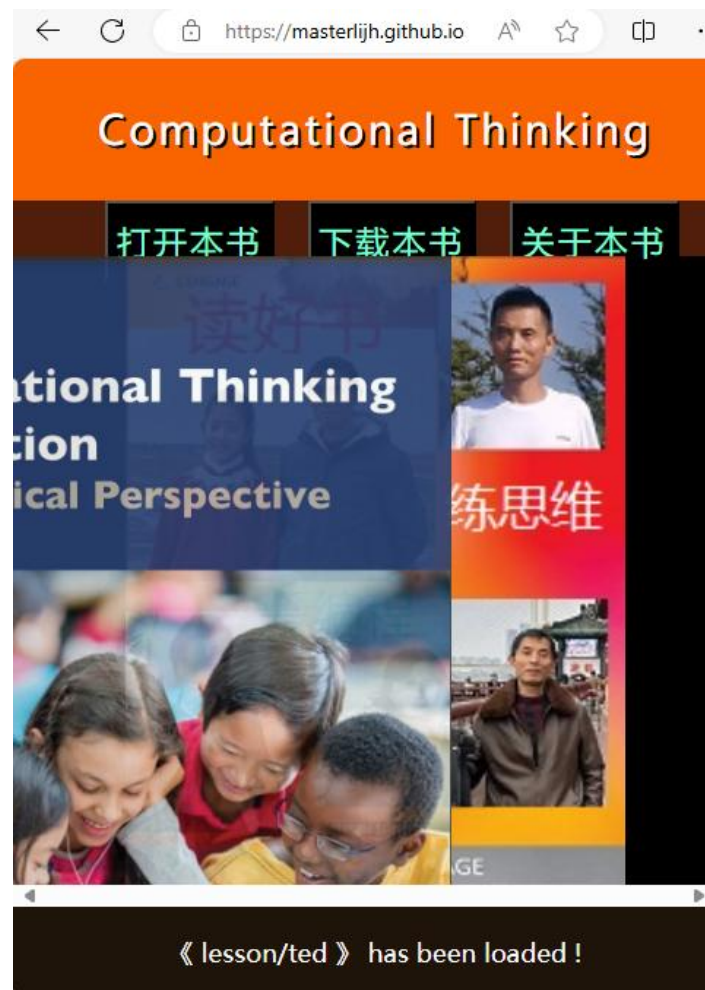


最后，GitHub 平台反馈：gitBash 和 gitHub 平台成功实现远程链接。



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

写作指导:

实质上任何章节都可以按三段论模式写，第一段是写研究的背景和目标，第二段是写你所开展的工作步骤和工作量，第三段是用了哪些方法和工作的结果或意义。

【摘要案例 1】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。

【摘要案例 2】：Web 技术以其跨操作系统平台的优势成为了广泛流行的软件开发手段，为了适应移动互联网时代软件项目的前端需求，本项目以 Web 客户端技术为研究学习内容，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量式开发模式，共做了 6 次项目迭代开发，每次迭代都经历了开发 4 个经典开发阶段（A:Analysis, D:Design, I: Implementation, T:Testing），以逐步求精的方式编写了本 UI 的应用程序。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。