

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）：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：梁港莎

学 号：20213618

指导教师：李建宏

2024 年 5 月 22 日

【摘要】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，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，采用了响应式设计编程，使得程序可以动态适用于当前 PC 端和移动端设备；此外，为了增加项目的趣味性和互动性，项目在在功能上以 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，具有很好的跨平台兼容性，不同设备下均可以实现相应的交互功能。其中，为鼠标和触屏设计了一个对象模型，代码得到了一定的优化，使得 javascript 代码更加模块化，结构清晰，提高了可维护性。本项目采用了增量式开发模式，以逐步求精的方式展开了六次代码的增量式重构（A:分析 D:设计 I:实现 T:测试），并采用了 git 工具进行版本管理，对每一次的项目设计进行提交、迭代，清晰记录下了项目开发的日志，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。

关键词：web 客户端技术；增量式开发模式；底层事件；响应式编程；git

第 1 章 前言

1.1 毕设任务分析

毕业设计作为我们本科学习阶段的重要实践和总结环节，要求我们将所学的计算机科学技术知识融会贯通，在实践层面应用这些核心课程的关键知识，架构自己感兴趣的技术路线，结合自己探求的问题形成软件需求，选择一个具有实际应用价值和研究意义的课题，然后有条理的系统落实分析问题、建立模型、软件设计、系统实施、测试调试等传统软件工程的全部的流程，开发出一个实用性软件项目。践行毕业设计，是大学理论的学习在实践层面做的一次综合演练总结，期间也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，再总结写毕业论文，从而在实践和理论二个维度训练学生专业的计算思维和工程维。

1.2 研学计划

➤ 技术路线研究

首先针对自己感兴趣的技术路线进行深入研究了解当前的技术现状、发展趋势以及应用前景通过查阅相关文献、参与技术论坛等方式，获取最新的技术信息和解决方案。

➤ 需求分析与系统设计

在确定了技术路线后，再进行详细的软件需求分析，明确软件的功能模块、性能要求以及用户界面设计。同时，根据需求分析结果进行系统设计，包括系统架构、数据库设计以及模块划分等。

➤ 编码实现与测试调试

在系统设计完成后，就可以开始编写代码实现软件功能。在编码过程中，尽量提高代码的优化性能，注重代码的可读性可维护性和性能优化。基本实现项目后，要进行单元测试、集成测试和系统测试等，确保软件应用的稳定性和逻辑正确性。在测试过程中，发现问题、解决问题，不断优化迭代项目。

➤ 文档写与论文总结

在软件开发完成后，进行详细的开发文档包括软件使用说明、技术手册和测试报告等。同时，对毕业设计过程进行总结，分析项目的成功之处和不足之处，对整个毕业设计过程进行系统的梳理和总结

1.3 研究方法

❖ 文献法

文献法作为一种重要的研究方法，在毕业设计中发挥着至关重要的作用，涉及广泛搜集、整理、分析和解读与研究主题相关的各类文献资料，从而全面、深入地了解研究领域的现状、发展趋势和存在的问题，一是专业书籍的学习与底层知识的获取；二是在线文档和社区论坛的文章，主要是 W3C 国际组织的 Web 标准和 Mozilla 基金会的 MDN 文档网站，他们在线上发表了大量公开文档，描绘了 web 的过去、现在、将来，获取到现下最先进的技术。W3C 主要以 Web 国际标准和最佳实践为主；三是通过知网看相关学术性的研究，为书写论文寻找一个好的模板模范和权威性的引用。

❖ 模型研究方式

一、运用结构化设计模型将软件系统划分为一系列独立的模块，每个模块负责完成特定的功能。通过模块之间的接口进行通信，实现系统的整体功能。结构化设计模型注重数据结构和算法，适用于处理复杂的数据处理和计算任务。

二、面向对象设计模型以对象作为软件系统的基本单元，强调对象之间的消息传递和协作。通过封装、继承和多态等机制。

第 2 章 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]。

Web 是一个文档的集合，被称为网页，它们由世界各地的计算机用户（在大部分时间内）共享。不同类型的网页可以做不同的事情，但至少，它们都能在电脑屏幕上显示内容。我们所说的“内容”是指文本、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，通过不同的工具实现不同类型的 Web 编程，Web 开发核心技术主要包括 HTML、CSS 和 JavaScript。HTML（超文本标记语言）用于描述网页的结构和内容；CSS（层叠样式表）用于控制网页的样式和布局；JavaScript 是一种脚本语言，用于实现网页的交互功能和动态效果。众所周知，HTML5、CSS 和 JavaScript 这三种技术是客户端 Web 编程的支柱，使用客户端 Web 编程，所有网页计算都在最终用户的计算机（客户端计算机）上执行^[3]。

在 Web 应用程序中，MVC 模式的应用尤为广泛，MVC 设计模式将软件分为模型(Model)、视图(View)和控制器(Controller)三部分。模型处理数据和业务逻辑，视图负责界面展示，控制器接收输入并协调模型与视图。HTML5、CSS 和 JavaScript 与 MVC 设计模式的结合体现了人类社会化大生产的分工智慧，使得前端开发变得更加高效、灵活和可维护。

2.2 项目的增量式迭代开发模式

软件生命周期中的开发过程包括四个阶段：分析、设计、实现和测试。对于开发过程，有几个模型。我们在这里讨论最常见的两种：瀑布模型和增量模型。

2.2.1 瀑布模型

软件开发过程中一个非常流行的模型是瀑布模型（如图 1）。在这个模型中，开发过程只流向一个方向。这意味着在上一个阶段完成之前才能启动一个阶段。例如，整个项目的分析阶段应在其设计阶段开始之前完成。整个设计阶段应在开始实施阶段之前完成。瀑布模型既有优点也有缺点。其中一个优点是，每个阶段都在下一个阶段开始之前完成。例如，在设计阶段工作的小组确切地知道该做什么，因为他们拥有分析阶段的完整结果。测试阶段可以测试整个系统，因为正在开发的整个系统已经准备好了。然而，瀑布模型的一个缺点是难以定位^[4]。

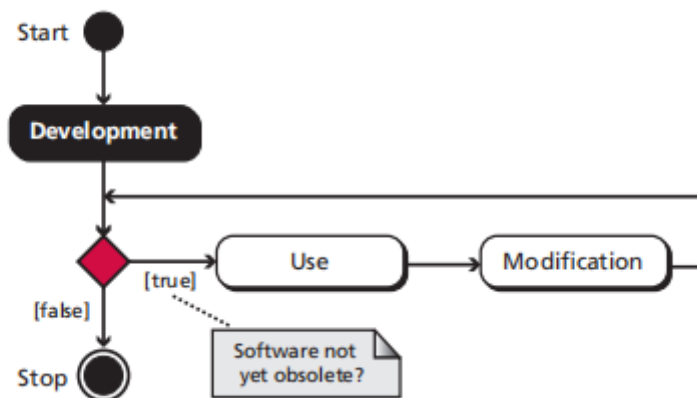


图 1. 瀑布模型

2.2.2 增量模型

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 2 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^[4]。

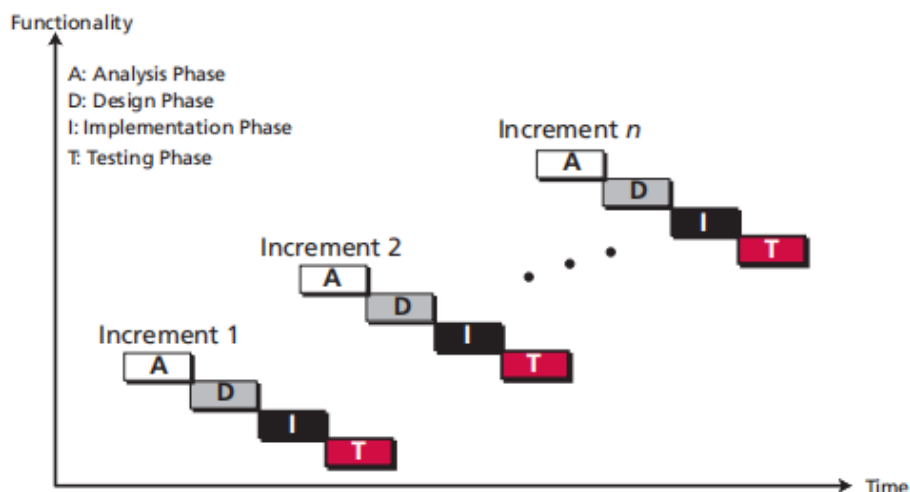


图 2. 增量模型

第 3 章 内容设计概要

3.1 分析与设计

三段式布局设计提供了清晰的结构和层次。本项目使用人们习惯的简洁的“三段论”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，然后展现的主要区域则是内容区，“内容为王”是项目必须坚守的理念，也是整个 UI 应用的主题，最后则是足部的附加信息。如上所述，头部、主体和底部各自承担不同的功能，使得网页内容得以有序展现。用户能够快速识别和定位所需信息，提升浏览效率，它不仅能够保证网页的整体性和一致性，还能提升用户体验和网站的可访问性。同时，这种布局方式也适用于不同屏幕尺寸和设备类型，具有较好的响应式设计特性，本次项目的用例图如下 3.1。

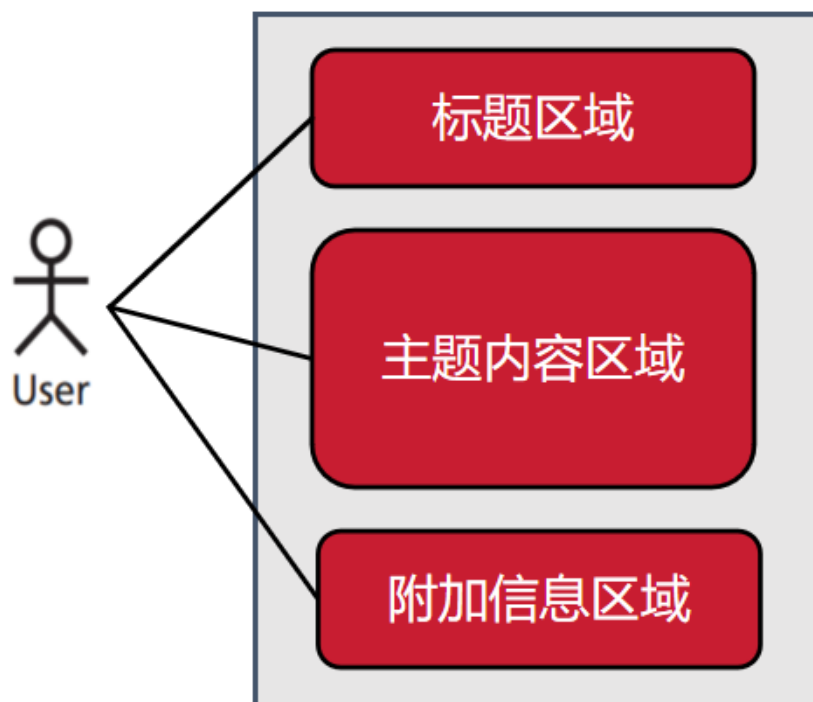


图 3.1. “三段式”用例图

3.2 代码的实现

从简洁的“三段论”方式开展内容设计，编写 1.1.html 对项目进行基本的内容设计，采用最简单的 HTML 页面结构，使用了 HTML5 的语义标签来构建网页的布局，代码的结构可以被分为三个主要部分，即头部（header）、主体（main）和底部（footer），是传统的三段式布局设计，其中 header 部分放了软件的标题，main 部分放入了软件的内容，footer 放了软件的动态反馈。

3.2.1 HTML 代码块

```
<header>
    《 计算机思维和语言学习 》
</header>
<main>

</main>
<footer>
    Copyright XXX 江西科技师范大学 2024-2025
```

</footer>

3.2.2 CSS 代码块

```
*{
  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 测试与运行（如图 3.2、3.3、3.4）



图 3.2（PC 端）

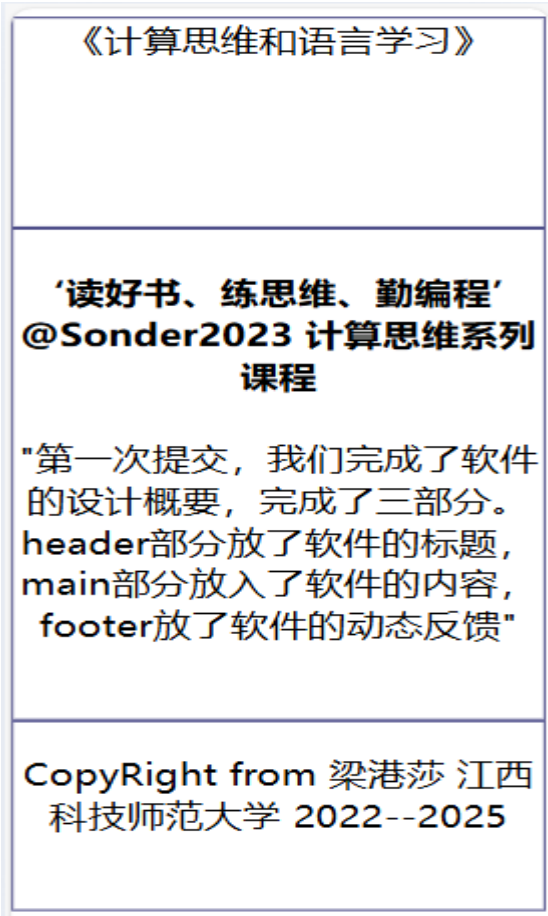


图 3.3（手机端）



图 3.4（平板）

通过手机扫描下方二维码实现手机端的观看：



WebUI_v1.0 移动端二维码

三段式布局设计提供了清晰的结构和层次。头部、主体和底部各自承担不同的功能，使得网页内容得以有序展现。用户能够快速识别和定位所需信息，提升浏览效率，它不仅能够保证网页的整体性和一致性，还能提升用户体验和网站的可访问性。同时，这种布局方式也适用于不同屏幕尺寸和设备类型，具有较好的响应式设计特性。

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，设置了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(保留每次的项目阶段，迭代 index.html)：

- ❖ `$mkdir webUI`
- ❖ `$cd webUI`
- ❖ `$mkdir webui`
- ❖ `$git init`
- ❖ `$git config user.name Sonder`
- ❖ `$git config user.email 3403994259@qq.com`
- ❖ `$touch index.html`
- ❖ `$touch mycss.cs`
- ❖ `$git add index.html`
- ❖ `$git add mycss.cs`
- ❖ `$git log`
- ❖ `cd webui`
- ❖ `$touch 1.1.html`
- ❖ `$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：

```
YOLO@LAPTOP-7JU4C3HI MINGW64 /d/webUI (master)
$ git commit -m 项目第一版本：“三段论”式的内容设计概要开发
[master (root-commit) fa813b5] 项目第一版本：“三段论”式的内容设计概要开发
2 files changed, 207 insertions(+)
create mode 100644 1.1.html
create mode 100644 index.html
```

项目代码仓库自此也开启了严肃的历史记录，我们可以输入日志命令查看，

❖ \$git log

```
YOLO@LAPTOP-7JU4C3HI MINGW64 /d/webUI (master)
$ git log
commit fa813b5c953d267d336a7a5f3bd8f709d182f16c (HEAD -> master)
Author: Sonder <3403994259@qq.com>
Date: Fri May 31 22:01:14 2024 +0800

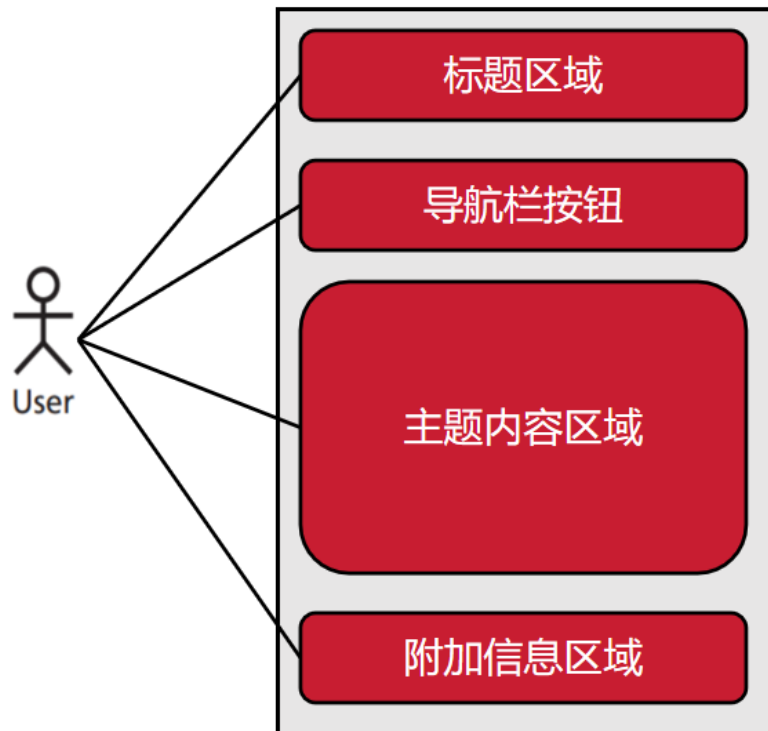
    项目第一版本：“三段论”式的内容设计概要开发
```

第4章 响应式设计和窄屏代码实现

4.1 响应性设计概述

响应式网页设计在不同的屏幕上可以自动调整页面布局的设计理念，不仅能够给用户带来更加流畅的体验，也直接增加了网站的点击率。响应式网页设计，具备多种终端浏览的流畅性和高清晰度，使得用户可以在不同设备上获得同样的体验，在增强用户体验感的同时，能够确保网页设计的质量及效果得到极为明显的提升，切实保障网页应用效果^[5]。

在分析响应式网页布局的应用时，首先要考虑到响应式网页设计要求网络页面能够兼顾 PC 端和移动端。页面布局可以根据设备屏幕的大小进行调整，并且确保页面内容不会变形，只是布局形式发生改变。为此，在使用时，响应式布局不仅要满足终端设备的屏幕显示要求，而且要达到网页设计目标中优秀的视觉效果，给用户提供良好的使用体验，在下一次使用时，能够提供更优质的页面浏览和搜索效果，确保在多种设备端都能以合理的布局展现给用户^[6]。本次项目用例图如下：



4.2 窄屏代码实现

本阶段设计了一个响应式设计的窄屏页面_1.2.html（如图 4.1、4.2），页面主要分为头部、导航、主体和底部，通过 JavaScript 动态计算屏幕宽度，并根据设备宽度调整页面字体大小和布局，实现了适配不同设备的效果，整体布局利用了百分比和相对长度单位，使页面元素可以自适应不同屏幕大小。

4.2.1 HTML 代码块

```
<header>
  <p id="book">
    Anime
    display@Sonder2023
  </p>
</header>
<nav>
  <button id="nav1" onclick="scrollToImage(0)">
    导航 1
  </button>
  <button id="nav2" onclick="scrollToImage(1)">
```

```

        导航 2
    </button>
    <button id="nav3" onclick="scrollToImage(2)">
        导航 3
    </button>
</nav>
<main id = 'main'>
    <div class="image-container" style="margin: 10px">
        <img src="" alt="Image 1">
        <p></p>
    </div>
    <div class="image-container" style="margin: 10px">
        <img src="" alt="Image 3">
        <p></p>
    </div>
    <div class="image-container" style="margin: 10px">
        <img src="" alt="Image 2">
        <p></p>
    </div>
</main>
<footer>
    <p id="statusInfo" >
        Copyright from 梁港莎 江西科技师范大学 2022--2025
    </p>
</footer>

```

4.2.2 CSS 代码块

本次 CSS 代码初次引入了 em 和 %，这是 CSS 语言中比较高阶的语法，em 和 % 单位是 CSS 中用于实现响应式设计的两个重要工具，它们都允许元素的尺寸和布局根据上下文环境动态调整。

```

<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>

```

4.2.3 JavaScript 代码块

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。

```

<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>

4.3 测试与运行（如图 4.1、4.2）

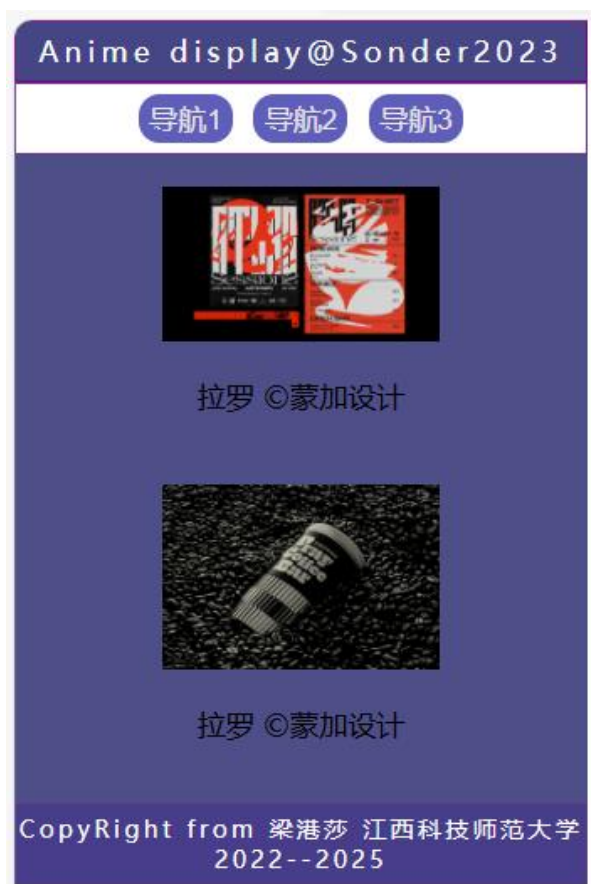


图 4.1（电脑端）



图 4.2（手机端）

扫描 QR 码



WebUI_v2.0 移动端二维码

4.4 项目的代码提交和版本管理

对第二次项目的代码进行提交，迭代 index.html、mycss.css 代码，增加 myjs.js 代码：

❖ `$cd webUI`

❖ `$touch myjs.js`

❖ `$git add myjs.js`

❖ `$git commit -m 项目第二版：响应式设计与窄屏代码的实现`

成功提交代码后，gitbash 的反馈如下所示：

```
YOLO@LAPTOP-7JU4C3HI MINGW64 /d/webUI (master)
$ git commit -m 项目第二版：响应式设计与窄屏代码实现
[master (root-commit) d226ef2] 项目第二版：响应式设计与窄屏代码实现
5 files changed, 289 insertions(+)
create mode 100644 index.html
create mode 100644 mycss.css
create mode 100644 myjs.js
create mode 100644 webui/1.1.html
create mode 100644 webui/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`

```
YOLO@LAPTOP-7JU4C3HI MINGW64 /d/webUI (master)
$ git log
commit d226ef2510f7b35d68c038b16a4c01dc14d29aa1 (HEAD -> master)
Author: Sonder <3403994259@qq.com>
Date: Wed Jun 5 16:04:50 2024 +0800
```

项目第二版：响应式设计与窄屏代码实现

第5章 宽屏和窄屏通用的响应式设计和代码实现

5.1 分析与设计

本阶段主要设计一个宽屏和窄屏通用的响应式页面，页面主要通过 CSS 和 JavaScript 实现了宽屏和窄屏的适配，使用百分比和 em 单位设置元素大小和字体，以保持布局的灵活性。其中 viewport 元信息确保了在移动设备上

的正确显示，JavaScript 动态调整字体大小和隐藏元素（如屏幕宽度小于 900px 时隐藏#aid），增强了在窄屏设备上的用户体验。

5.2 代码实现

5.2.1 HTML 代码块

首先对页面布局进行设计，页面主要分为头部、主体和底部，与上个阶段相比添加了一个额外的区域，用于显示用户键盘响应区，如图代码块 5-1。

```
<header>
  <p id="book">
    《现代电影赏析》
  </p>
</header>
<main id="main" >
  <div id="bookface" >
  </div>
</main>
<footer>
  Copyright from 梁港莎 江西科技师范大学 2022--2025
</footer>
<div id="aid" >
  <p>用户键盘响应区</p>
  <div class="image-container" style="margin: 10px">
    <img src="" alt="Image 1">
    <p>Howl's Moving Castle</p>
    <div style="font-size: 25px;" contentEditable="true">
      <h3>影片赏析:</h3>
    </div>
  </div>
</div>
</div>
```

代码块 5-1

5.2.2 CSS 代码块

本次 CSS 代码（如图代码块 5-2）中依旧运用了 em 和%单位用于实现响应式设计，它们都允许元素的尺寸和布局根据上下文环境动态调整，其中也使用了绝对定位（Absolute Positioning）使得元素精确控制在页面上，这对于创建复杂的布局和设计非常有用。

```

<style>
  *{
    text-align: center;
    box-sizing: border-box ;
  }

  header{
    border: 2px solid blue;
    height: 15%;
    font-size: 1.66em;

  }
  main{
    border: 2px solid blue;
    height: 70%;
    font-size: 1.2em;

  }
  footer{
    border: 2px solid blue;
    height: 5%;
  }
  body{
    position:relative ;
  }
  #aid{
    position: absolute;
    border: 3px solid blue;
    top: 0.5em;
    left: 600px;
  }
  #bookface{
    width: 80%;
    height: 80%;
    border:1px solid red;
    background-color: blanchedalmond;
    margin:auto;
  }
</style>

```

代码块 5-2

5.2.3 JavaScript 代码块

在本次的 JavaScript 中，依旧运用了上一个阶段项目的 UI 对象，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，与上个阶段不同的是，代码中增加了判断语句，通过判断窗口宽度，当窗口宽度小于 900px 时，隐藏#aid 元素，还定义了 mouse 对象，用于跟踪鼠标状态和位置，并且为#bookface 元素添加了鼠标事件监听器，用于跟踪鼠标按下、移动和离开事件（增加的代码如图代码块 5-3）。

```
<script>
    .....
    if(window.innerWidth < 900){
        $("#aid").style.display='none';
    }
    $("#aid").style.width=window.innerWidth - UI.appWidth - 2*baseFont + 'px';
    $("#aid").style.height= document.body.clientHeight + '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";
    });
</script>
```

</script>

代码块 5-3

5.3 测试与运行（如图 5.1、5.2、5.3）



图 5.1（PC 端）



图 5.2（手机端 iPhone）

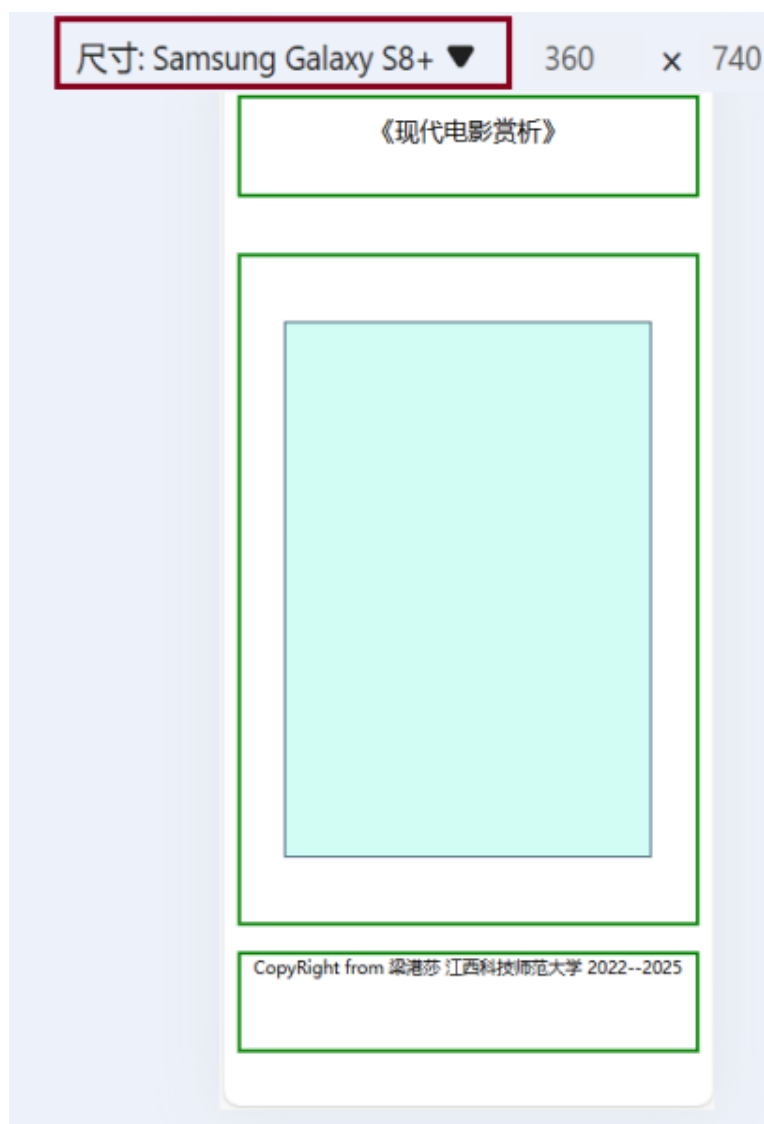


图 5.2（手机端 Samsung）

扫描 QR 码



WebUI_v3.0 移动端二维码

5.4 项目的代码提交和版本管理

对第三次项目的代码进行提交，迭代 index.html、mycss.css、myjs.js 代码：

❖ **\$git commit -m** 项目第三版：窄屏和宽屏通用的响应式设计代码实现

成功提交代码后，gitbash 的反馈如下所示：

```
YOLO@LAPTOP-7JU4C3HI MINGW64 /d/webUI (master)
$ git commit -m 项目第三版：窄屏和宽屏通用的响应式设计代码实现
[master d6a5bf6] 项目第三版：窄屏和宽屏通用的响应式设计代码实现
4 files changed, 331 insertions(+), 120 deletions(-)
rewrite index.html (84%)
rewrite mycss.css (82%)
rewrite myjs.js (92%)
create mode 100644 webui/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**

```
YOLO@LAPTOP-7JU4C3HI MINGW64 /d/webUI (master)
$ git log
commit d6a5bf6755085f2736f03167b959cb9a8bc82aa8 (HEAD -> master)
Author: Sonder <3403994259@qq.com>
Date: Wed Jun 5 16:27:46 2024 +0800

    项目第三版：窄屏和宽屏通用的响应式设计代码实现

commit d226ef2510f7b35d68c038b16a4c01dc14d29aa1
Author: Sonder <3403994259@qq.com>
Date: Wed Jun 5 16:04:50 2024 +0800

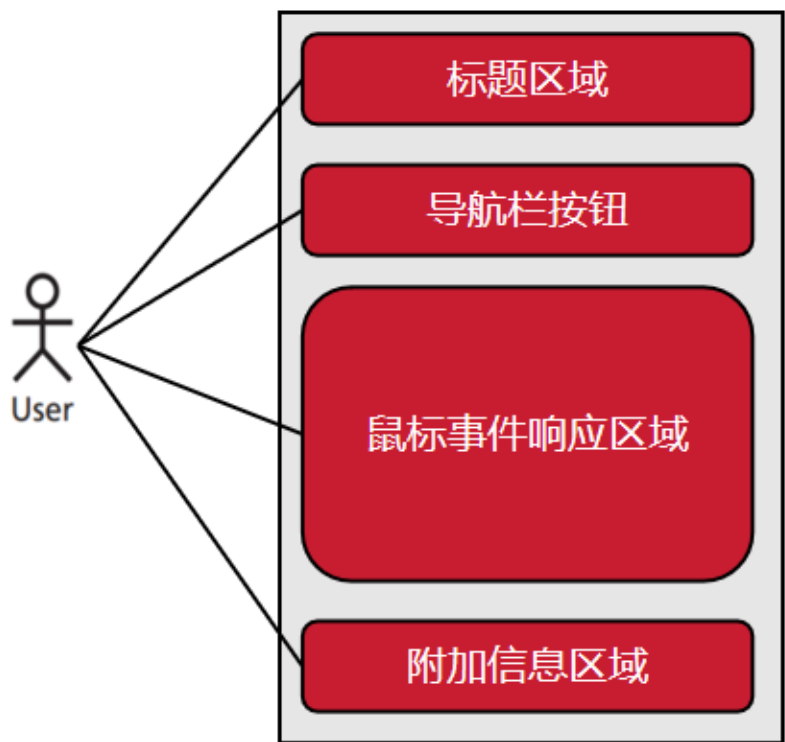
    项目第二版：响应式设计和窄屏代码实现
```

第6章 个性化交互 UI 初步——鼠标模型的设计

6.1 分析与设计

本次项目主要是对鼠标事件在页面中的应用而展开，鼠标事件在 Web 开发中扮演着至关重要的角色，其增强了网页的交互性，使得用户能够通过鼠标操作与网页内容进行更直观、更丰富的互动，鼠标事件可以创建一个动态和

响应式的与用户界面，用户可以通过鼠标点击、悬停、拖拽等操作来浏览网页，极大增加了用户的体验感。此外，鼠标事件可以在不同的操作系统和设备上进行操作，使得 Web 应用能够跨平台工作，本次项目设计添加了几个主要的鼠标事件，如 mousedown、mouseup、mouseout 等事件，对比之前的项目拥有更多的可操作性和交互性，以下为本次项目的用例图，基于下方用例图展开对本次项目功能的设计。



6.2 代码的实现

6.2.1 HTML 代码块

对页面布局进行设计，页面保持为头部、导航、主体和底部的基本布局，其中主体部分为鼠标事件的响应，在主体区域内拖动鼠标进行事件的发生，如图代码块 6-1。

```
<header>
  <p style="color: #0c2a46">
    Slide Image
  </p>
</header>
<nav>
  <button>向前</button>
```

```

        <button>向后</button>
        <button>其他</button>
</nav>

<main id="main">
    <div id="bookface">
        //鼠标事件的响应区域
    </div>
</main>

<footer>

    Copyright 梁港莎 江西科技师范大学 2024--2025

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

```

代码块 6-1

6.2.2 CSS 代码块

本次 CSS 代码（如图代码块 6-2）中依旧可以实现响应式设计，在鼠标响应区域添加了背景图片，增加本次项目的趣味性与美观性，还对按钮进行了一些外观的设计。

```

<style>
*{
    *****

}
header{
    *****

}
nav{
    *****

}
main{

```

```

        *****
    }

    footer{
        *****
    }
    body{
        position: relative;
    }
    button{
        font-size: 18px; /* 指定字体大小 */
        width: 100px; /* 保持按钮宽度 */
        height: 40px; /* 保持按钮高度 */
        color: #fafafa;
        background-color: #0c2a46;
        border: none;
        cursor: pointer;
    }
    #aid{
        position: absolute;
        border: 3px solid #4f4f88;
        top:0px;
        left:600px;
    }
    #bookface{
        position: absolute;
        width: 80%;
        height: 80%;
        background-color: #ded2ee;
        background-image:
url("https://cdn.iconfinder.com/data/family/previews/preview/checkers.p
ng?indexed=1702900638");
        left:7% ;
        top: 7% ;
    }
</style>

```

代码块 6-2

6.2.3 JavaScript 代码块

在本次的 JavaScript 中，依旧运用了上一个阶段项目的 UI 对象，与上个阶段不同的是，代码中增加了多个鼠标事件，其中，鼠标按下事件（mousedown）可以记录下鼠标按下的坐标，当用户释放鼠标开始后触发鼠标释放事件（mouseup），鼠标在响应区域移动时触发鼠标事件（mousemove），用户可以在水平方向上移动该响应区域，即可实现图片的滑动，最后，鼠标离开响应区域时触发鼠标事件（mouseout），刷新页面，移动的区域回到原位。本次代码基本实现了面向用户的交互式页面，用户控制鼠标，即可在相应的区域对页面的元素进行控制，提升用户的体验感。（增加的代码如图代码块 6-3）。

```
<script>
.....
if(window.innerWidth < 900){
    $("aid").style.display='none';
}
$("aid").style.width=window.innerWidth - UI.appWidth - 2*baseFont + 'px';
$("aid").style.height= document.body.clientHeight + 'px';
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= "";
        if(Math.abs(mouse.deltaX) > 100){
            $("bookface").textContent += " " ;
        }else{
            $("bookface").textContent += " " ;
            $("bookface").style.left = '7%' ;
        }
    });
}
```

```

    }

});
$("bookface").addEventListener("mouseout",function(ev){
    ev.preventDefault();
    mouse.isDown=false;

    $("bookface").textContent= "";
    if(Math.abs(mouse.deltaX) > 100){
        $("bookface").textContent += " " ;
    }else{
        $("bookface").textContent += " " ;
        $("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= "" + mouse.deltaX +"";
        $('bookface').style.left = mouse.deltaX + 'px' ;
    }
});
</script>

```

代码块 6-3

6.3 测试与运行（如图 6.1、6.2、6.3）

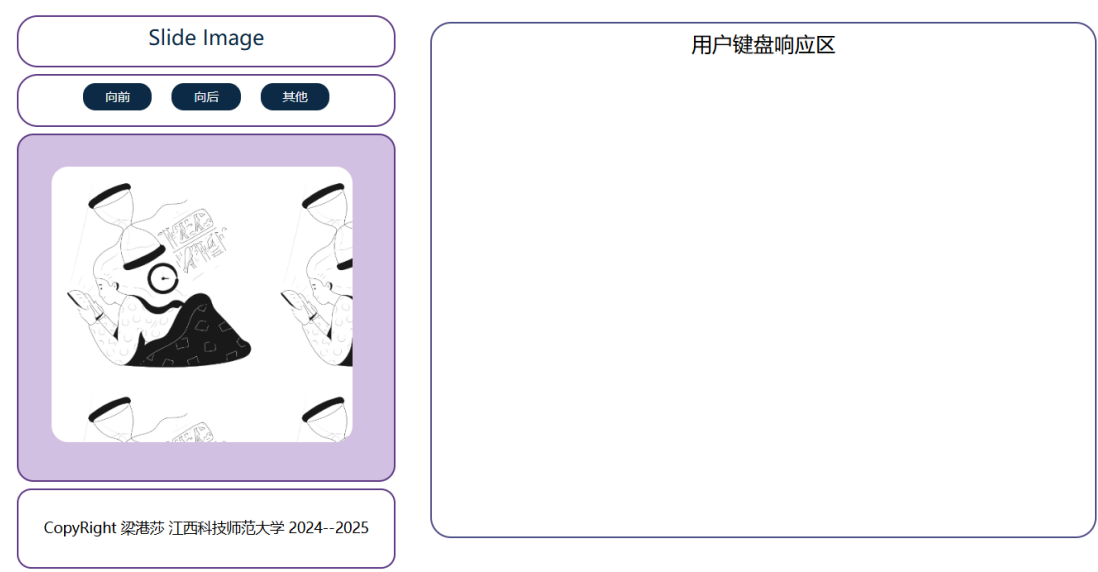


图 6.1（PC 端）



图 6.2 (手机端 iPhone)



图 6.3 (手机端 Samsung)

扫描 QR 码



WebUI_v4.0 移动端二维码

6.4 项目的代码提交和版本管理

对第四次项目的代码进行提交，迭代更新 index.html、mycss.css、myjs.js 代码：

❖ **\$git commit -m 项目第四版：个性化 UI 设计中的鼠标模型**

成功提交代码后，gitbash 的反馈如下所示：

```
YOLO@LAPTOP-7JU4C3HI MINGW64 /d/webUI (master)
$ git commit -m 项目第四版：个性化UI设计中的鼠标模型
[master f234ad9] 项目第四版：个性化UI设计中的鼠标模型
1 file changed, 197 insertions(+)
create mode 100644 webui/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**

```
YOLO@LAPTOP-7JU4C3HI MINGW64 /d/webUI (master)
$ git log
commit f234ad9a0bc777582ca1c4b13856c50a40fb0b8a (HEAD -> master)
Author: Sonder <3403994259@qq.com>
Date:   Wed Jun 5 16:35:39 2024 +0800

    项目第四版：个性化UI设计中的鼠标模型

commit d6a5bf6755085f2736f03167b959cb9a8bc82aa8
Author: Sonder <3403994259@qq.com>
Date:   Wed Jun 5 16:27:46 2024 +0800

    项目第三版：窄屏和宽屏通用的响应式设计代码实现

commit d226ef2510f7b35d68c038b16a4c01dc14d29aa1
Author: Sonder <3403994259@qq.com>
Date:   Wed Jun 5 16:04:50 2024 +0800

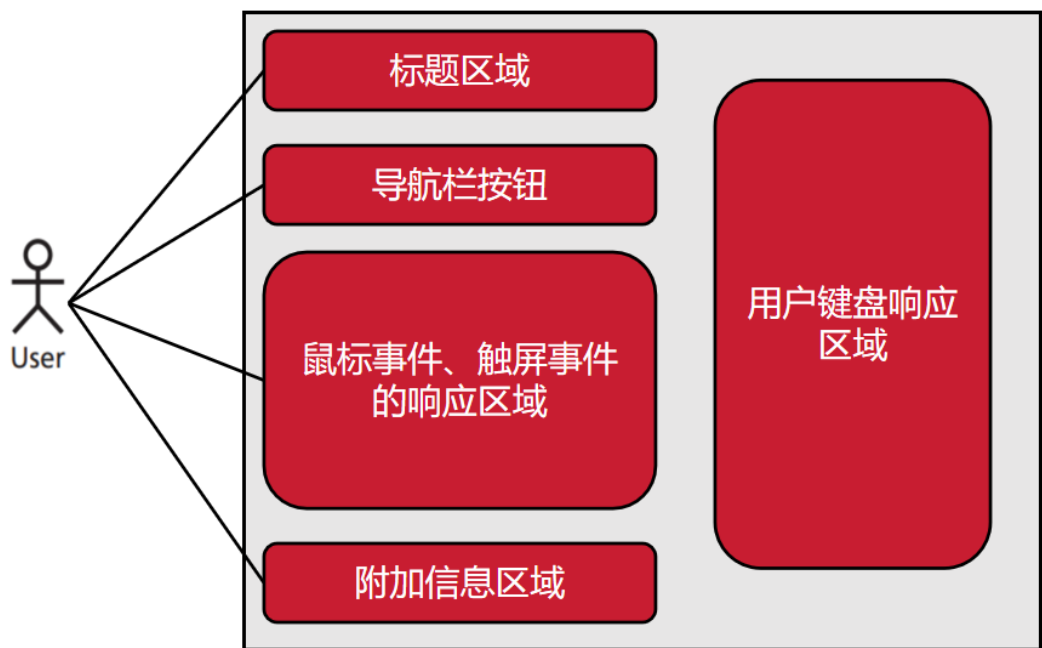
    项目第二版：响应式设计和窄屏代码实现
```

第7章 通用的 UI 设计，用一套代码同时为触屏和鼠标建模

7.1 分析与设计

本阶段的项目在原有的基础上添加触屏和键盘事件，结合鼠标事件创建一个

更加多样的交互体验，为了提高开发效率和满足不同设备的要求，主要对 JavaScript 代码进行相应的优化，尝试运用一套代码同时为触屏和鼠标建模，运用统一的事件处理函数，减少了为鼠标和触屏分别编写和维护代码的工作量，使得代码更加模块化，提高了可维护性，代码结构也更加清晰可读，此外，用一套代码控制触屏和鼠标事件的 UI 设计具有很好地跨平台兼容性，响应性强，同时满足不同设备的要求，PC 端用户使用鼠标事件进行交互，手机端、平板端的用户使用触屏事件完成响应。下方为本阶段的用例图。



7.2 代码的实现

7.2.1 HTML 代码块

对页面布局进行设计，页面保持为头部、导航、主体和底部的基本布局，其中主体部分为鼠标事件的响应和触屏事件的响应，在主体区域内拖动鼠标进行事件的发生，如图代码块 7-1 (与上一阶段项目的布局相同)。

```
<header>
  <p style="color: #0c2a46">
    Slide Image
  </p>
</header>
<nav>
  <button>向前</button>
```

```

        <button>向后</button>
        <button>其他</button>
</nav>

<main id="main">
    <div id="bookface">
        //鼠标事件、触屏事件的响应区域
    </div>
</main>

<footer>

    Copyright 梁港莎 江西科技师范大学 2024--2025

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

```

代码块 7-1

7.2.2 CSS 代码块

本次 CSS 代码（如图代码块 7-2）中依旧可以实现响应式设计，在鼠标响应区域、触屏区域添加了背景图片。

```

<style>
*{
    *****
}
header{
    *****
}
nav{
    *****
}
main{
    *****
}
footer{
    *****
}

```

```

    }
    body{
        *****
    }
    button{
        *****
    }
    #aid{
        *****
    }
    #bookface{
        *****
    }
}
</style>

```

代码块 7-2

7.2.3 JavaScript 代码块

在本次的 JavaScript 代码中，依旧运用了上一个阶段项目的 UI 对象，与上个阶段不同的是，代码中增加了触屏事件和键盘事件，其中，触屏和鼠标事件运用了同一套代码，统一地使用了三个主要的事件处理函数 handleBegin、handleEnd 和 handleMoving，分别在鼠标点击或触屏触摸开始交互时触发、释放鼠标按钮或手指离开触屏时触发、移动鼠标或触屏时触发，这样代码更加地模块化结构化，此外，通过 ev.touches 的存在来判断是触屏事件还是鼠标事件。与上一阶段代码的 JavaScript 代码相比，本次的事件监听器是统一添加的，避免了重复代码，并且使得事件监听器的添加过程更加集中和有序。最后，在键盘响应区域实现了任意键的显示。

```

{ //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= "";
        if(Math.abs(Pointer.deltaX) > 100){
            $("#bookface").textContent += " " ;
        }else{
            $("#bookface").textContent += " " ;
            $("#bookface").style.left = '7%' ;
        }
    }else{

        $("#bookface").textContent= "";
        if(Math.abs(Pointer.deltaX) > 100){
            $("#bookface").textContent += " " ;
        }else{
            $("#bookface").textContent += " " ;
            $("#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);
let count = 0;
document.body.addEventListener("keypress", function(ev){
    if(count === 30){
        $("keyboard").textContent += "\n";
        count = 0;
    } else {
        $("keyboard").textContent += ev.key;
        count++;
    }
});
}

```

代码块 7.3

7.3 测试与运行（如图 7.1、7.2、7.3）

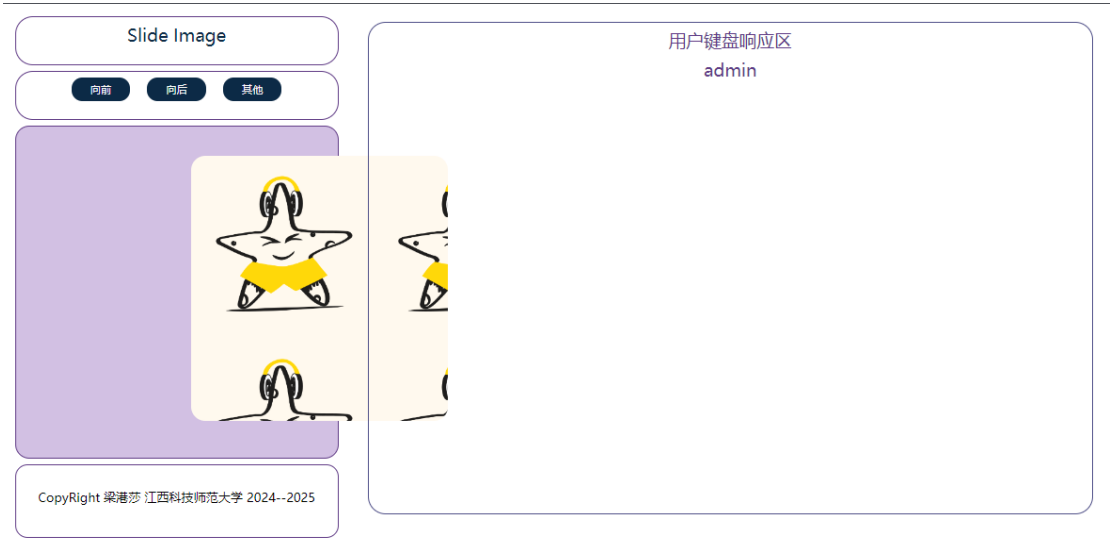


图 7.1（PC 端）

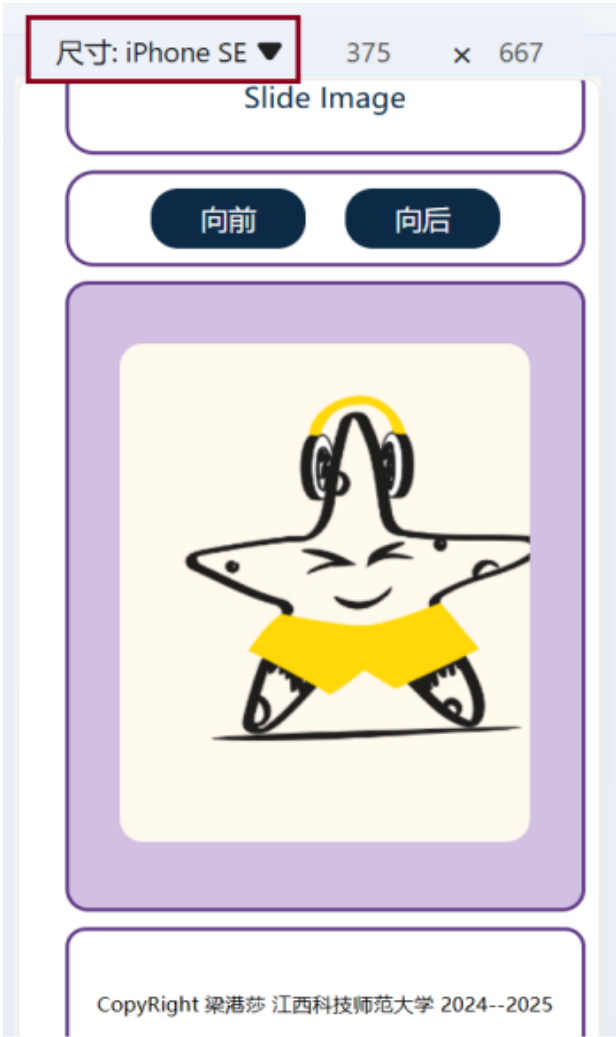


图 7.2（手机端 iPhone）



图 7.3（手机端 Samsung）

通过手机扫描下方二维码实现手机端的观看：

扫描 QR 码



7.4 项目的代码提交和版本管理

对第五次项目的代码进行提交，迭代 index.html、mycss.css、myjs.js 代码：

❖ **\$git commit -m** 项目第五版：通用的 UI 设计，用一套代码为触屏和鼠标建模

成功提交代码后，gitbash 的反馈如下所示：

```
YOLO@LAPTOP-7JU4C3HI MINGW64 /d/webUI (master)
$ git commit -m 项目第五版：通用的UI设计，用一套代码为触屏和鼠标建模
[master a93559a] 项目第五版：通用的UI设计，用一套代码为触屏和鼠标建模
4 files changed, 389 insertions(+), 92 deletions(-)
rewrite index.html (71%)
create mode 100644 webui/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**

```
YOLO@LAPTOP-7JU4C3HI MINGW64 /d/webUI (master)
$ git log
commit a93559a5db5dc6a4dc7baa778de115ac21128d48 (HEAD -> master)
Author: Sonder <3403994259@qq.com>
Date:   Wed Jun 5 16:45:19 2024 +0800

    项目第五版：通用的UI设计，用一套代码为触屏和鼠标建模

commit f234ad9a0bc777582ca1c4b13856c50a40fb0b8a
Author: Sonder <3403994259@qq.com>
Date:   Wed Jun 5 16:35:39 2024 +0800

    项目第四版：个性化UI设计中的鼠标模型

commit d6a5bf6755085f2736f03167b959cb9a8bc82aa8
Author: Sonder <3403994259@qq.com>
Date:   Wed Jun 5 16:27:46 2024 +0800

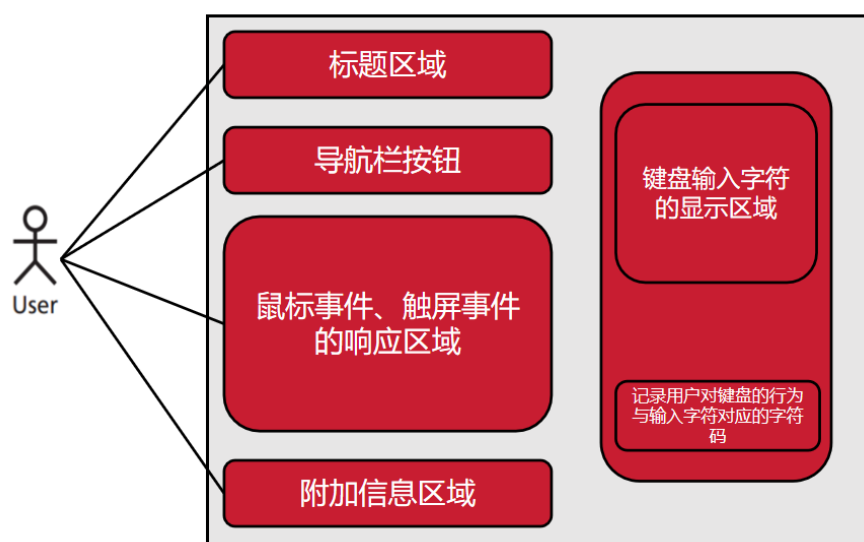
    项目第三版：窄屏和宽屏通用的响应式设计代码实现

commit d226ef2510f7b35d68c038b16a4c01dc14d29aa1
Author: Sonder <3403994259@qq.com>
Date:   Wed Jun 5 16:04:50 2024 +0800
```


第8章 UI 的个性化键盘交互控制的设计开发

8.1 分析与设计

本阶段项目为最后一个版本的设计，主要对用户的键盘响应区进行改进，应用 keydown 和 keyup 键盘底层事件进行代码优化，创建了一个简单的键盘控制工具，实时显示用户按下键盘的字符及其字符码，对用户的按下键盘和松开键盘的动作进行实时记录，可以对输入的字符进行删除的操作，该监听键盘事件又更进一步地丰富用户的交互体验，以下为本次项目的用例图。



8.2 代码实现

8.2.1 HTML 代码块

本次布局主要是将用户键盘响应区域细分为头部的标题区域、中间部分区域显示用户按下键盘的字符、尾部区域实时记录下用户对键盘的动作及按下字符的字符码（如下代码块）。

```
<header>
  <p style="color: #0c2a46">
    Slide Image
  </p>
</header>
```

```

<nav>
    <button>向前</button>
    <button>向后</button>
    <button>其他</button>
</nav>

<main id="main">
    <div id="bookface">

        </div>
</main>

<footer>

    Copyright 梁港莎 江西科技师范大学 2024--2025

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

```

8.2.2 CSS 代码块

与之前的样式设计大致相同，主要是对用户键盘响应区的各个区域进行位置设计，位置设计如下。

```

#status{
    position: absolute;
    text-align: center;
    right: 0; left: 0;
    bottom: 0;
}

```

8.2.3 JavaScript 代码块

本次 JavaScript 代码添加了 keydown 事件和 keyup 事件，keydown 事件实时捕捉到按键动作，可以重复触发，如果用户持续按住某个键，keydown 事件会连续触发多次，而当用户释放键盘上的任意键时触发 keyup 事件，通

过使用 `keydown` 和 `keyup` 事件，其中，使用了调用 `preventDefault()` 方法，使得用户按下 `Backspace`（退格键）时实现字符的删除功能。总的来说，代码实现了一个简单的键盘输入监控功能，可以实时显示用户按下的键，并在按下 `Backspace` 时删除最近输入的字符。本次项目成功创建出了响应用户键盘操作的动态和交互式 Web 应用。主要的实现代码块如下：

```
$("#body").addEventListener("keydown", function(ev) {
    let k = ev.key;
    let c = ev.keyCode;
    // 检查是否按下了 Backspace 键
    if (k === "Backspace") {
        ev.preventDefault(); // 阻止默认行为
        let keyboardText = $("#keyboard").textContent;
        // 移除最后一个字符
        if (keyboardText.length > 0) {
            $("#keyboard").textContent = keyboardText.slice(0,
-1);
        }
    }
    // 更新按键状态显示
    $("#status").textContent = "您已按键 : " + k + " , " + "字符编
码 : " + c;
});

// 监听键盘释放事件，用于更新按键状态显示
$("#body").addEventListener("keyup", function(ev) {
    let k = ev.key;
    let c = ev.keyCode;
    $("#status").textContent = "松开按键 : " + k + " , " + "字符编
码 : " + c;
});

// 监听按键按下事件，用于实时显示按键输入
$("#body").addEventListener("keypress", function(ev) {
    if (ev.key !== "Enter" && ev.key !== "Backspace") { // 排除
Enter 和 Backspace
        $("#keyboard").textContent += ev.key;
    }
});
```

});

8.3 测试与运行（如图 8.1、8.2）



图 8.1（PC 端）



图 8.2 (手机端 iphone)

通过手机扫描下方二维码实现手机端的观看：



WebUI_v6.0 移动端二维码

8.4 项目的代码提交和版本管理

对第六次项目的代码进行提交，迭代 index.html、mycss.css、myjs.js 代码：

❖ **\$git commit -m** 项目第六版：UI 的个性化键盘控制——应用 keydown 和 keyup 键盘底层事件

成功提交代码后，gitbash 的反馈如下所示：

```
YOLO@LAPTOP-7JU4C3HI MINGW64 /d/webUI (master)
$ git commit -m 项目第六版：UI的个性化键盘控制——应用keydown和keyup键盘底层事件
[master 31f7fed] 项目第六版：UI的个性化键盘控制——应用keydown和keyup键盘底层事件
4 files changed, 296 insertions(+), 33 deletions(-)
create mode 100644 webui/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**

```
YOLO@LAPTOP-7JU4C3HI MINGW64 /d/webUI (master)
$ git log
commit 31f7fed5850ba705701996bf2b6f9f87a5227137 (HEAD -> master)
Author: Sonder <3403994259@qq.com>
Date:   Wed Jun 5 16:56:14 2024 +0800

    项目第六版：UI的个性化键盘控制--应用keydown和keyup键盘底层事件

commit a93559a5db5dc6a4dc7baa778de115ac21128d48
Author: Sonder <3403994259@qq.com>
Date:   Wed Jun 5 16:45:19 2024 +0800

    项目第五版：通用的UI设计，用一套代码为触屏和鼠标建模

commit f234ad9a0bc777582ca1c4b13856c50a40fb0b8a
Author: Sonder <3403994259@qq.com>
Date:   Wed Jun 5 16:35:39 2024 +0800

    项目第四版：个性化UI设计中的鼠标模型

commit d6a5bf6755085f2736f03167b959cb9a8bc82aa8
Author: Sonder <3403994259@qq.com>
Date:   Wed Jun 5 16:27:46 2024 +0800
```

第 9 章 本项目的高质量代码的编写

随着互联网技术的发展，Web 程序的应用也日趋广泛，JavaScript 所承担的作用也不只是设计之初那样处理简单的 DOM 交互，更多的是需要为图形处理，物理引擎和虚拟现实等计算密集型操作提供支持。然而 JavaScript 以易用性为主的设计理念导致其性能上无法高效的应对繁重的处理需求，特别是在计算资源资源有限的平台。因此如何能够有效的提升 JavaScript 代码的执行效率，这对于网络应用程序的发展至关重要^[7]。

本项目就是主要对 JavaScript 代码进行相应的优化，尝试运用一套代码同时为触屏和鼠标建模，运用统一的事件处理函数，减少了为鼠标和触屏分别编写和维护代码的工作量，使得代码更加模块化，提高了可维护性，代码结构也更加清晰可读，此外，用一套代码控制触屏和鼠标事件的 UI 设计具有很好地跨平台兼容性，响应性强，同时满足不同设备的要求。

第 10 章 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。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^[7]。

10.2 设置本地仓库和远程代码仓库的连接

10.2.1 设置本地仓库

❖ `$mkdir webUI`

//创建 webUI 文件夹

❖ `$cd webUI`

❖ `$mkdir webui`

❖ `$git init`

//在当前目录 webUI 中初始化一个新的 Git 仓库。这会创建一个新的 .git 目录，Git 用来存储仓库的元数据。

❖ `$git config user.name Sonde`

//配置 Git 仓库的用户名称为 Sonder。

❖ `$git config user.email 3403994259@qq.com`

//配置 Git 仓库的用户电子邮件地址

10.2.2 将本地仓库上传至 GitHub 的远程仓库上

生成密钥：

```

YOLO@LAPTOP-7JU4C3HI MINGW64 ~
$ ssh-keygen -t rsa -C "3403994259@qq.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/YOLO/.ssh/id_rsa):
Created directory '/c/Users/YOLO/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/YOLO/.ssh/id_rsa
Your public key has been saved in /c/Users/YOLO/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:jmH8MEnea8/VSK8wtDDnngzJ5WVCd0D3dYEFBMuAivo 3403994259@qq.com
The key's randomart image is:
+---[RSA 3072]-----+
|      . 000.=o+      |
|    E o  o+.oo      |
|   +   . .oo o      |
|  = o  . . . .      |
|   O S = +          |
|  . O @ * +         |
|  . O * o o         |
|   * = .            |
|   * .              |
+---[SHA256]-----+

```

添加密钥到 GitHub:

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys



key

SHA256:jmH8MEnea8/VSK8wtDDnngzJ5WVCd0D3dYEFBMuAivo

Added on Jun 11, 2024

Last used within the last week — Read/write

Delete

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

与远程仓库建立链接并上传

在 GitHub 上建立一个新仓库后，进入本地 webUI 项目的文件夹，通过下面的命令把本地代码仓库与远程建立密钥链接

```

$ git remote set-url origin git@github.com:Sonder2023/2024-UI.git
$ git push -u origin main

```

成功后，本地仓库的代码上传到了在 GitHub 创建远程仓库中

main	1 Branch	0 Tags	Go to file	Add file	Code
Sonder2023 项目第六版: UI的个性化键盘控制——应用keydown和keyup键盘底层事件 ✓ 8b3e8d1 · 12 minutes ago 14 Commits					
webui	项目第五版: 通用的UI设计, 用一套代码为触屏和鼠标建模				15 minutes ago
index.html	Update index.html				45 minutes ago
mycss.css	项目第六版: UI的个性化键盘控制——应用keydown和keyup...				last week
myjs.js	项目第六版: UI的个性化键盘控制——应用keydown和keyup...				last week

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：



通过生成网站二维码，扫描二维码实现手机端访问：



参考文献

- [1]W3C.W3C's history.W3C Community.[EB/OL].<https://www.w3.org/about/>.2023.12.20
- [2]John Dean,PhD.Web programming with HTML5,CSS,and JavaScript[M].Jones&Bartlett Learning,LLC.2019: 2
- [3]John Dean,PhD.Web programming with HTML5,CSS,and JavaScript[M]. Jones & Bartlett Learning,LLC. 2019:xi
- [4]Behrouz Forouzan. Foundations of Computer Science[M](4th Edition). Cengage Learning EMEA,2018: 274--275
- [5]直敏,高天哲,孙杨.响应式布局在网页设计中的应用[J].无线互联科技,2022,19(15):150-152.
- [6]刘申.响应式网页设计中栅格系统的构建方法研究[J].信息与电脑(理论版),2021,33(09):159-161.
- [7]薛超.基于 WebAssembly 的 JavaScript 性能优化方案研究与实现[D].西北大学,2019.