

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：郭昕奕

学 号：20213645

指导教师：李健宏

2024 年 6 月 13 日

目录

题目（中文）：基于 Web 客户端技术的个性化 UI 设计和实现	1
基于 Web 客户端技术的个性化 UI 设计和实现	4
第 1 章 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	6
1.3.1 文献法	6
1.3.2 模型研究法	6
第 2 章 技术总结和文献综述	7
2.1 概述	7
2.2 项目的增量式迭代开发模式	8
第 3 章 内容设计概要	11
3.1 分析和设计	11
3.2 项目的实现和编程	12
3.3 项目的运行和测试	13
3.4 项目的代码提交和版本管理	14
第 4 章 移动互联时代的响应式设计和窄屏代码实现	15
4.1 分析与设计	15
4.2 项目的实现与编程	16
4.3 项目的运行和测试	17
4.4 项目的代码提交和版本管理	18
第 5 章 适用移动互联时代的响应式设计	20
5.1 分析和设计	20
5.2 项目的实现和编程	21
5.3 项目的运行和测试	21
5.4 项目的代码提交和版本管理	22
第 6 章 个性化 UI 设计中鼠标模型	24

6.1 分析和设计	24
6.2 项目的实现和编程	25
6.3 项目的运行和测试	26
6.4 项目的代码提交和版本管理	27
第 7 章 通用的 UI 设计	28
7.1 分析与设计	28
7.2 编程与实现	29
7.3 测试与运行	31
7.4 代码提交	32
第 8 章 UI 的个性化键盘控制	32
8.1 分析与设计	32
8.2 编程与实现	33
8.3 测试与运行	35
8.3 代码提交	36
第 9 章 本项目中的高质量代码	36
第 10 章 用 gitBash 工具管理本项目的 http 服务器	38
10.1 经典 Bash 工具介绍	38
10.2 设置本地仓库和远程代码仓库的链接	39
参考文献	40

基于 Web 客户端技术的个性化 UI 设计和实现

【摘要】在当今这个数字化、信息化的时代，Web 技术因其无与伦比的跨平台性，已逐渐成为软件开发领域的主流选择。无论是大型企业级的复杂系统，还是个人开发者的小型项目，Web 技术都以其便捷性、可访问性和广泛的用户基础，赢得了众多开发者的青睐。我们的项目正是以 Web 客户端技术为核心，深入探索了 HTML、CSS 和 JavaScript 这三大基石级的技术，并不仅仅是停留在表面的知识学习，而是深入到这些技术的实际应用中，进行了细致的研究和实践。在这个过程中，我们特别参考了 MDN（Mozilla Developer Network）社区，这个汇聚了全球开发者智慧和经验的平台，为我们提供了大量宝贵的资源和实践指导。通过不懈的努力和团队的协作，我们成功地将这些技术整合在一起，结合我们本科阶段所学的课程知识，构建出了一个具有高度响应性和个性化的用户界面。这个界面不仅能够在传统的 PC 端设备上完美运行，更能够自适应各种移动设备，无论是手机、平板还是其他便携式设备，都能够为用户提供流畅、舒适的体验。为了实现这一目标，我们采用了多种技术手段。其中，DOM（文档对象模型）技术和事件驱动模式是我们最为核心的工具。通过这些技术，我们能够实现用户界面的动态更新和交互反馈，支持用户通过鼠标、触屏和键盘等多种方式进行操作。无论用户是在哪种设备上，都能够感受到一致的、无缝的交互体验。

关键词：Web 客户端技术；WebUI 设计；DOM

第 1 章 前言

我们作为计算机科学技术专业的本科生，在即将完成学业之际，的确很有必要开发一个专业的作品，来回顾本学科专业学习的知识系统，梳理课程体系中最核心的东西。在该项目设计中，涉及的有关核心课程的理论包括面向对象的程序设计语言、数据结构和算法、操作系统、软件工程等。项目设计的内涵就是大学理论的学习在实践层面做一次综合的演练和总结，期间配合一些当下最新的流行技术，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，再总结出论文。深刻理解计算机系统对于我们开发者而言，是非常重要的，这也是我们不同于其他专业的特色，在其他专业严重看来，我们是计算机专业的，我们对于计算机系统的理解一定不是浮于表面的，而是尽量地接近计算机的本质，对于任何技术的理解则是能接近技术的底层和基本原理。

计算机科学与技术专业研究的内容非常丰富，可以说是一个大学科，CS 专业的方向大致分为以下三种，一是计算机硬件体系架构的研究，这种方向接近微电子数字电路和芯片研究，二是系统软件体系和架构管理的研究，这种方向需要深入操作系统的体系架构和网络，三是应用软件开发方向的研究。不管哪个方向，都需要经历发现问题、分析问题、解决问题这个过程。

1.1 毕设任务分析

毕业论文的内涵就是大学理论的学习在实践层面做一次综合演练和总结，期间也需要配合学习当前的一些新技术，在已形成自己个人对于计算机软硬件体系的系统而专业的理解后，再进行总结撰写论文，这就是毕业论文的内涵。毕设是本科教育的重要环节，其综合性和个性化都很强，对于提升本科生的各项能力意义很大。我们可以从个人成长的高度和意义来做好这个环节的方方面面，为了具有时代感，我结合国家实现中国式现代化的目标来写，从教育人才的高度来理解教育部和学校的这项规划。

还是引用导师李教授的观点：毕设任务开始后，本科生要尽早从自身需要出发主动规划，让设计/论文成为自己主动学习和自我教育的最后一战。围绕本专业的目标，选择好自己的方向和技术路线，在导师的指导下，成长为本专业的“四有人才”：1、具有现代工程思维的工程师；2、掌握现代数字工具的人；3、具有国际视野能自主学习的学习者；4、兼具人文情怀全面发展的人。第1，2点素质主要支撑毕设的开发，而第3，4点主要支撑论文协作，我想此次毕设任重而道远，为了少走弯路，亲手实践构建代码，广泛阅读查阅文献文档，总结撰写文档是正确走好这段路程的行为范式。

1.2 研学计划

整个设计将分为两阶段完成，首先就是选择一条自己感兴趣的实践路线，将核心的技术加以整理学习，以导师的案例项目为参考，理解好各个技术之间的关系以及它们在项目中的分工，同时在这个过程中提升自己写代码的能力。以上就是第一阶段。

第二阶段，一般按照软件工程的标准来规范开发，结合自己的问题做出定义

和分析，设计出一套合适的技术解决方案，然后按照设计好的解决方案编写出相关的代码实现技术部署，最后进行调试、测试和调优。其实毕设和个人开发者很类似，项目的设计和具体实现都没有经验丰富的开发团队，很多时候是为了提高效率，方案设计的优化和写代码的具体部署其实是交替进行的，前者是工程师落实微观和细节层面，后者则是设计师的工作，确保宏观层面的设计不偏离需求。在开发期间可以产生大量的开发文档，对这些文档做出总结，再结合本专业的理论就可以形成自己的论文，进而加深本科期间学习的理论的真是体会，从实践升华到理论，再用理论实现最佳实践。

1.3 研究方法

1.3.1 文献法

对于文献法的描述可以用一句话表达，即“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”，对于我们 Web 应用方向的学生来说，在自己代码能力成长的任何阶段，我们都无法离开文献法，然而承载知识的媒体形式非常丰富，包括书籍、在线文档、社区论坛、期刊等等。

我的毕设的资料来源有下面几类：

第一就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，二者分别站在科学家和工程师的两个层次，描绘了 web 的过去、现在和将来。W3C 主要以 web 国际标准和最佳实践为主，而 MDN 则偏向于 Web 的最佳实践和各层面应用的教学与讨论。

第二就是国人用的最多的知网平台，在这里可以学习论文的规范，了解毕设选题他人理论上的研究和探索，对于本科生而言可提供的理论指导和展示学术的规范。

1.3.2 模型研究法

对于写代码的本科生，必须擅长使用的另一种研究方法就是“模型研究法”。这个研究方法非常具体，也很有意思，其给人的快乐甚至可以与打游戏对比。比

如承载我们 Web 应用的台式机、笔记本、手机、平板，传递在线信息要用到的互联网、服务器，沟通硬件和我们的代码之间的操作系统、浏览器、代码编辑器、编译器，这些软硬件对象，对我们而言，都值得从写代码的角度去研究，我们笼统地称它们为对象，这些对象最终会在我们大脑中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，最后写出代码来，这种行为本质上就是先在思维上“建模”，再用 OOP 语言表述出来。

在 OOP 分析和开发过程中，我在毕设中试图解决的问题，也被定义成为了各级各种模型。模型研究只是更为抽象，与具体的计算机语言无关，在毕设中我也尝试使用国际标准 UML (Unified Modeling Language) 语言来建立抽象模型。我感觉采用 UML 模型研究法和面向对象的程序设计方法的目标是一致的，只是在不同层面分析表述问题而已。[5] 因此采用模型研究法，我的毕设再用例设计采用了类似的 UML 对问题建模，景观 UML 比较抽象，设计准确有一定难度，而使用 OOP 程序对画好的模型开展程序设计则更为具体直观，通过熟悉的 OOP 语言和代码运行环境运行和调试模型，我们甚至可以倒推出模型设计的问题和缺陷。我个人的观点是对于本科生而言，可能直接写代码建立模型研究模型，代码跑通后，再利用 UML 语言绘制模型，作为代码的文档资料则更合理。毕竟本科生在思维上还很稚嫩，无法比拟高级程序员乃至系统架构师那样的抽象能力和丰富的经验。

第 2 章 技术总结和文献综述

2.1 概述

应用软件开发的方向内涵非常丰富，也是非常庞杂的，仅仅是表达代码的现代高级计算机语言就不下十几种，而且每种都是博大精深并且独具特色的。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 Web 编程

我们先从 Web 的简要描述开始，Web 是 World Wide Web 的缩写。大多数人会说“Web”而不是“World Wide Web”，我们也将遵循这一惯例。Web 是由一系列文档组成的集合，这些文档被称为网页，它们（在大多数情况下）被世界各地的计算机用户共享。不同类型的网页有不同的功能，但最基本的是，它们都在计算机屏幕上显示内容。这里的“内容”指的是文本、图片以及用户输入机制，如文本框和按钮^[2]。

Web 编程是一个广泛的领域，由不同类型的 Web 编程实现，由不同的工具执行。所有工具都使用核心语言 HTML 工作，因此几乎所有的 Web 编程书籍都在某种程度上描述了 HTML。这本教科书深入介绍了 HTML5、CSS 和 JavaScript。这三种技术被称为客户端 Web 编程的支柱。通过客户端 Web 编程，所有网页计算都在最终用户的计算机（客户端计算机）上执行。^[3]

Web 应用的程序设计体系由三大语言有机组成：HTML，CSS， JavaScript。这三大语言的组合也体现了人类社会化大生产分工的智慧，可以看作用三套相对独立体系实现了对一个信息系统的描述和控制，可以总结为：HTML 用来描述结构（Structure）、CSS 用来描述外表（presentation）、Javascript 用来描述行为（Behavior）[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。

2.3 项目的增量式迭代开发模式

为了在互联网上更加方便且直观地展示我们所收集到的一些艺术作品的相关信息(包括图片以及作者，创作日期等内容)，我计划搭建一个开源的个性化展示网站。同时，为了增加网站的可读性和美观性，我们首先给这些信息进行了分

类，我将页面划分为三个主要部分包括标题区、内容区、附加信息区等。

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。

软件工程视角中，软件生命周期中主要包括开发和维护两项工作，其中在我们的本科毕业设计中仅涉及开发过程。软件的开发过程包括四个阶段：分析 (analysis)、设计 (design)、实现 (implementation) 和测试 (testing)，下文简称为 ADIT。而开发过程，有经典的两种模型：瀑布模型和增量模型。

2.3.1 瀑布模型

在瀑布模型中，开发过程是非常严格且单向的，在 ADIT 的四个阶段流程中，只有在上一阶段完成之后才能启动下一阶段。比如我们在实施过程中发现了设计过程中的问题，是无法推翻设计过程的，这一点对我们普通开发者和小团队是非常不方便的，^[4]因此我的毕业设计就采用了另一个开发模式，即增量式模式。

在这个模型里，项目启动时，项目团队专注于定义产品和项目的总体范围，然后制定产品（及相关可交付成果）交付计划，接着通过各阶段来执行计划。应该仔细管理项目范围变更。如果有新增范围，则需要重新计划和正式确认。对于经常变化的项目而言，瀑布模型毫无价值。

以下情况优先选择这种生命周期：项目需求明确、充分了解拟交付的产品、有厚实的行业实践基础、或者整批一次性交付产品有利于干系人。例如开发一个软件项目时，如果采用这个模型的话，一般将软件开发分为可行性分析（计划）、需求分析、软件设计（概要设计、详细设计）、编码（含单元测试）、测试、运行维护等几个阶段，如下图所示。

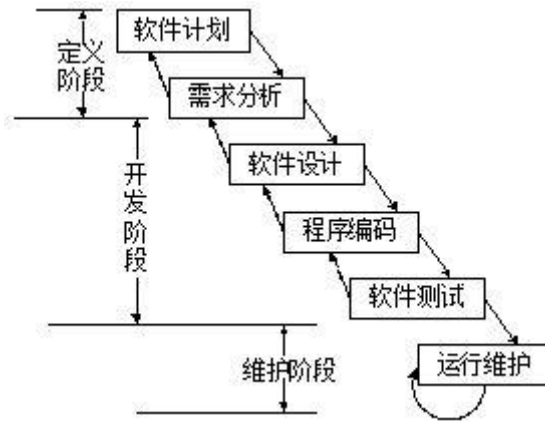


图 2.1 瀑布模型

2.3.2 增量模型

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]。

增量模型是把待开发的软件系统模块化，将每个模块作为一个增量组件，从而分批次地分析、设计、编码和测试这些增量组件。运用增量模型的软件开发过程是递增式的过程。相对于瀑布模型而言，采用增量模型进行开发，开发人员不需要一次性地把整个软件产品提交给用户，而是可以分批次进行提交如下图 2-2 所示：

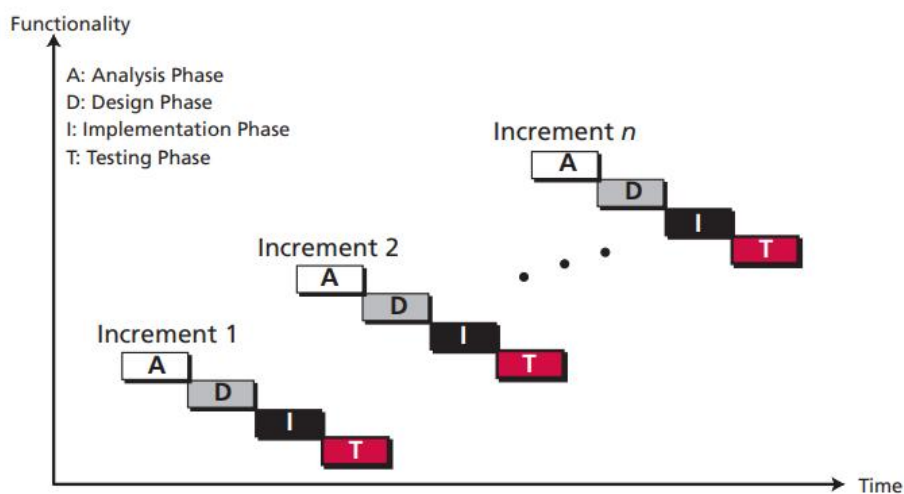


图 2-2 增量模型

第 3 章 内容设计概要

3.1 分析和设计

人们获取信息的途径是多样且不断变化的，在一些固定场所比如家中或教室，我们更倾向于使用电脑来进行学习和工作。但是，在一些变化的场景如出行时，我们却更倾向于使用手机和平板等设备来进行信息的快速查看。那现在我们的网站仅仅在电脑端实现了较为舒适的布局，如果在手机端呢？它可能就错位了！因此，我们需要一个同时适用于电脑端和手机端的网站，以确保无论在哪一种设备上，用户都能获得良好的使用体验。为了在互联网上更加方便且直观地展示我们收集到的一些相关信息，我们打算搭建一个可视化的网站，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 3-1 用例图所示：

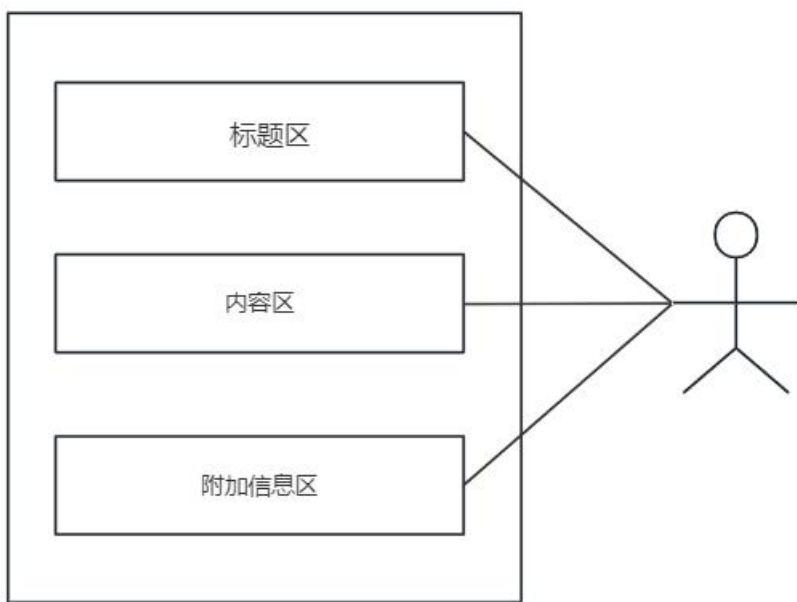
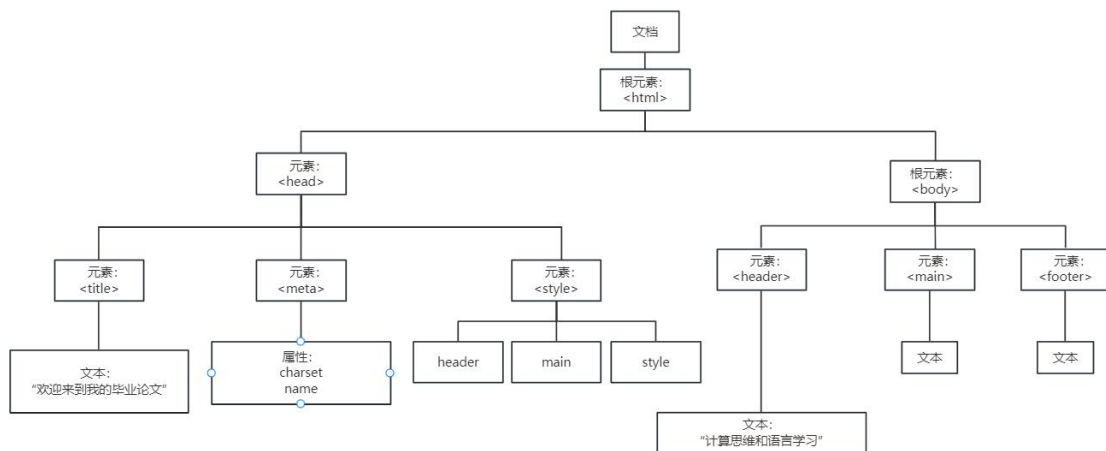


图 3-1 用例图

项目 dom 树图如下：



3.2 项目的实现和编程

为了让页面能够自动适应不同设备的屏幕尺寸，使其呈现出良好的自适应效果，在宏观上，我利用 JavaScript 动态获取设备窗口的宽度和高度信息，以便于根据不同设备的特性，动态调整页面元素的大小。代码块如下：

```
<title>欢迎来到我的毕业论文</title>
</head>
```

```

<body>
  <header>
    计算思维和语言学习
  </header>
  <main>
    Web 编程是一个广泛的领域，由不同类型的 Web 编程实现，由不同的工具执行。所有工具都使用核心语言 HTML 工作，因此几乎所有的 Web 编程书籍都在某种程度上描述了 HTML。这本教科书深入介绍了 HTML5、CSS 和 JavaScript。这三种技术被称为客户端 Web 编程的支柱。通过客户端 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。
  </main>
  <footer>
    Copyright from 郭昕奕 江西科技师范大学 2022--2025
  </footer>

```

页面样式代码如下：

```

<style>
  header{
    border:2px solid blue;
    height: 200px;
  }
  main{
    border:2px solid blue;
    height: 200px;
  }
  footer{
    border:2px solid blue;
    height: 200px;
  }
</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 3-2 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 3-3 的二维码，运行测试本项目的第一次开发的阶段性效果。



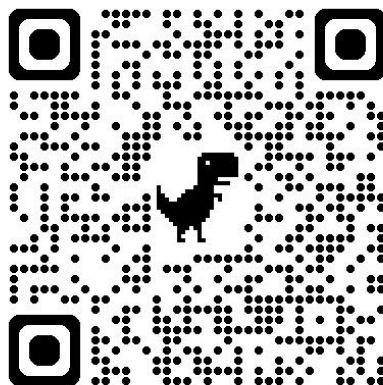
图 3-2 PC 端运行效果图



图 3-3 移动端运行效果

第一次提交，我们完成了软件的设计概要，完成了三个部分。header 部分放了软件的标题，main 部分放了软件的内容，footer 放了软件的动态反馈。

项目地址二维码如下图所示：



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 郭昕奕  
$ git config user.email 2509009907@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：

```
qcymyg@DESKTOP-G8R5382 MINGW64 /webUI (master)  
$ git commit -m 项目第一版：“三段论”式论文内容设计概要开发  
[master (root-commit) 6fff01b] 项目第一版：“三段论”式论文内容设计概要开发  
2 files changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 index.html  
create mode 100644 myCss.css  
qcymyg@DESKTOP-G8R5382 MINGW64 /webUI (master)  
$ |
```

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：

```
qcymyg@DESKTOP-G8R5382 MINGW64 /webUI (master)  
$ git log  
commit 6fff01b67db57948f6f0af7071a5a1edee191843 (HEAD -> master)  
Author: 郭昕奕 <2509009907@qq.com>  
Date:   Wed Jun 12 08:52:08 2024 +0800  
  
    项目第一版：“三段论”式论文内容设计概要开发  
qcymyg@DESKTOP-G8R5382 MINGW64 /webUI (master)  
$ |
```

第 4 章 移动互联时代的响应式设计和窄屏代码实现

4.1 分析与设计

在人机交互的过程中，鼠标是不可或缺的关键一环，所以我们在设计个性化用户界面时，探究了一下 U 设计中的鼠标模型，特别是关注了鼠标在计算机中所呈现的坐标信息。我们希望当对电脑固定区域进行鼠标点击事件时，页面会显示响应的鼠标的坐标信息。为了增加网站的可读性和美观性，我们首先给这些信息

进行了分类，我将页面划分为三个主要部分包括标题区、内容区、附加信息区等，如图 4-1 所示。

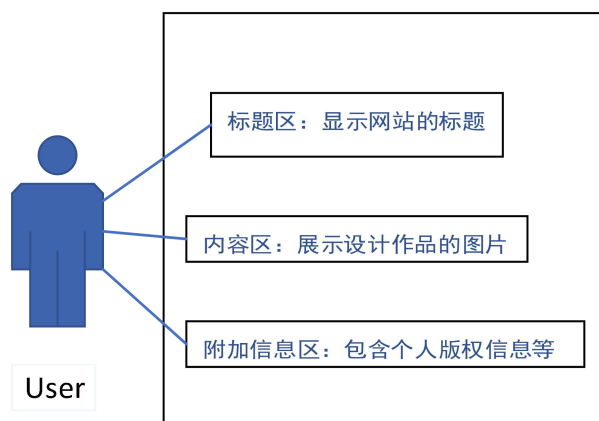


图 4-1 内容划分图

4.2 项目的实现与编程

为了使项目看上去更加的美观以及直接，我在此次的项目中加入了一些样式以及内容，页面上添加了三个导航栏起到分割效果。通过 CSS 技术，根据设备的屏幕尺寸，分辨率和方向等特性，实现了自动调整页面布局、字体大小以及图片尺寸。在窄屏设备上，页面布局应尽可能简洁明了，避免过多的层次和复杂的元素。可以使用单列布局或卡片式布局等方式来展示内容。项目代码块如下：

```
<title>Welcome to myCTApp</title>
</head>
<body>
  <header>
    <p id="book">
      剑桥雅思系列课程@gxy
    </p>
  </header>
  <nav>
    <button>
      导航 1
    </button>
    <button>
      导航 2
    </button>
    <button>
```



```

        导航 3
    </button>
</nav>
<main>
    
</main>
<footer>
    <p id = "statusInfo">
CopyRight from 郭昕奕 江西科技师范大学 2022--2025
    </p>
</footer>
<script>
    var UI = {};
    UI.deviceWidth = window.innerHeight;
    UI.deviceWidth = window.innerWidth;
    document.body.style.height = UI.innerHeight + 'px';
    document.body.style.fontSize = UI.deviceWidth / 22 + 'px';
</script>

```

4.3 项目的运行和测试

项目的测试至少需要通过两类终端，但由于篇幅有限，本文只给出 Edge 以及手机端运行后的效果图，如下图 4-1，图 4-2 所示：

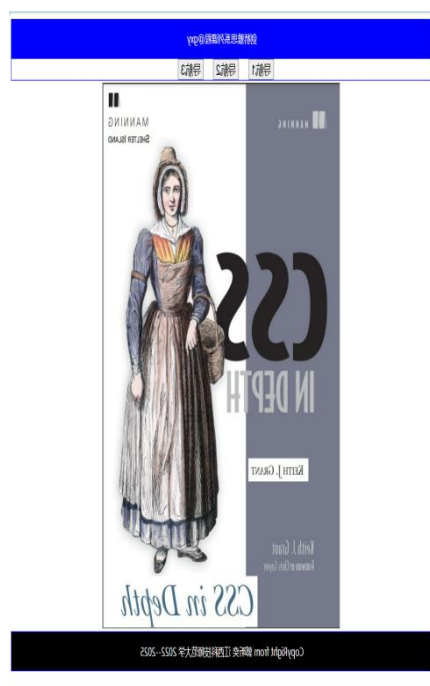


图 4-2 Edge PC 端运行效果

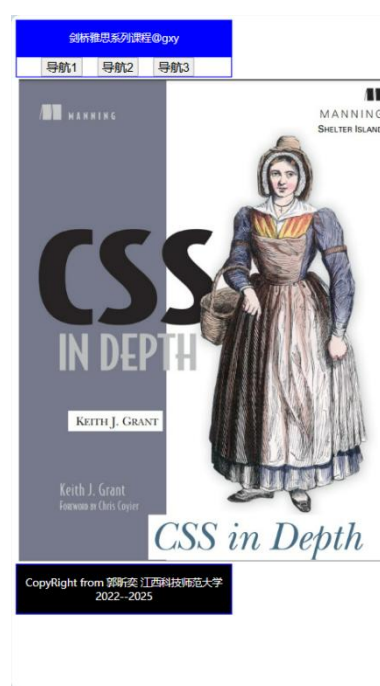


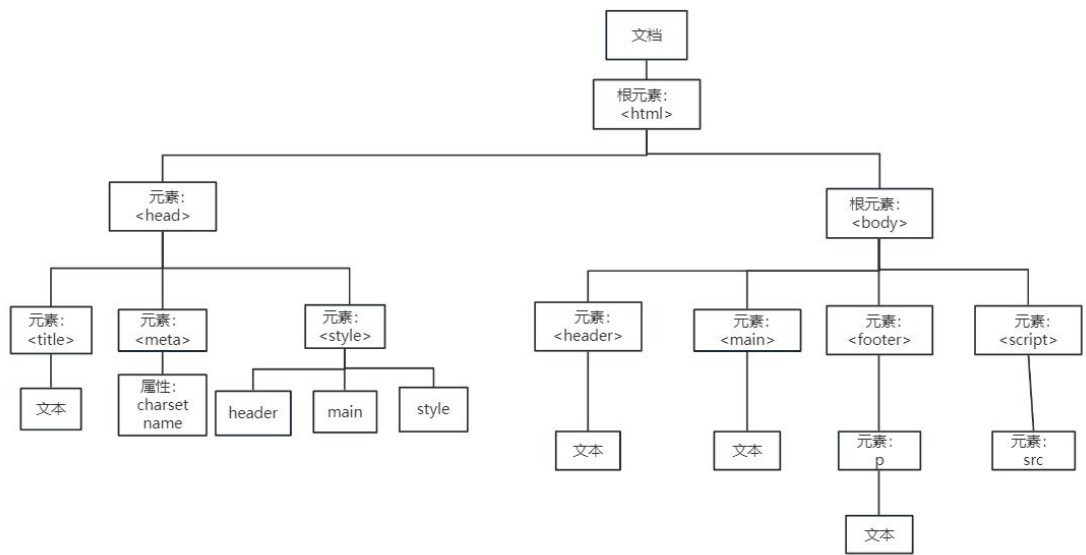
图 4-3 手机端运行效果

本次代码进行了应用程序的响应式设计：1. 为四个区域设置了占整个用户界面的比例（以高度为准）；2. 用 javascript 获取了用户屏幕的宽度和高度，并依据需要进行了计算，设置了基础字体的大小 3. 在不同的区域根据内容的重要程度设定了该区域字体的相对大小；4. 改变了颜色和对色，对主区域的背景图做了优化外观的设计；5. 总之，初步为用户的应用程序做了响应式设计并部署了代码。

项目地址二维码如下图所示：



项目 dom 树图如下所示：



4.4 项目的代码提交和版本管理

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。

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。如以下代码块所示：

```
<style>
  header{
    border:2px solid blue;
    height: 10%;
    background-color: blue;
    color: white;
  }
  body{
    font-size: 26px ;
    text-align: center;
  }
  nav{
    border: 2px solid blue;
    height: 5%;
  }
  main{
    height: 75%;
    background-image: url(../lesson/CS.jpg);
    background: no-repeat;
    background-size: cover;
    background-position: center;
  }
  footer{
    border:2px solid blue;
    height: 10%;
    background-color: black;
    color: white;
  }
  p#id{
    font-size: 1.8em;
    letter-spacing: 0.2em;
  }
  button{
    font-size: 1.2em;
    margin-right: 1em;
  }
</style>
```

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。

```
<script>
  var UI = {};
  UI.deviceWidth = window.innerHeight;
  UI.deviceWidth = window.innerWidth;
  document.body.style.height = UI.innerHeight + 'px';
  document.body.style.fontSize = UI.deviceWidth / 22 + 'px';
</script>
```

接着，我使用 gitbash 工具将以上项目提交：

```
git config user.name 郭昕奕
git config user.email 2509009907@qq.com
touch index.html.html myCss.css
git add index.html myCss.css
git commit -m 项目第二版：移动互联网时代的响应式设计和窄屏实现
```

输入 git log 查看提交日志

```
qcymyg@DESKTOP-G8R5382 MINGW64 /webUI (master)
$ git log
commit 16e1d652e29a23a4086a7a06be64b8b0f6e488ef (HEAD -> master)
Author: 郭昕奕 <2509009907@qq.com>
Date: Wed Jun 12 14:49:58 2024 +0800

    项目第二版：移动互联网时代的响应式设计和窄屏实现

commit 6fff01b67db57948f6f0af7071a5a1edee191843
Author: 郭昕奕 <2509009907@qq.com>
Date: Wed Jun 12 08:52:08 2024 +0800

    项目第一版：“三段论”式论文内容设计概要开发
```

第 5 章 适用移动互联时代的响应式设计

5.1 分析和设计

移动互联时代的用户终端多样性体现在不同设备类型、屏幕尺寸、分辨率、操作系统和浏览器等方面。用户可能使用智能手机、平板电脑、笔记本电脑、台式电脑以及甚至智能手表等设备来访问网站或应用程序，并且在此次的项目更新中还初步添加了鼠标交互和键盘交互。

响应式布局旨在确保网页能够在不同尺寸和分辨率的显示设备上有效显示，而不必为每个设备类型创建单独的网页版本。设计师通过制定一般布局指导原则，让浏览器根据具体设备的特性（如屏幕宽度和分辨率）自行选择如何呈现页面内容。这样，网页内容（如段落、图片等）可以自动调整以适应不同的设备和浏览器，确保用户无论使用何种设备都能获得良好的浏览体验。

5.2 项目的实现和编程

本次设计，我添加了鼠标响应区以及用户键盘的响应区，当用户移动到鼠标响应区时，会显示鼠标正在移动以及鼠标已经离开等提示，而当用户敲击键盘时，响应区也会提示出相应的字母以及编码。

5.3 项目的运行和测试

本次设计项目运行与测试结果如下如 5-2，图 5-2 所示：

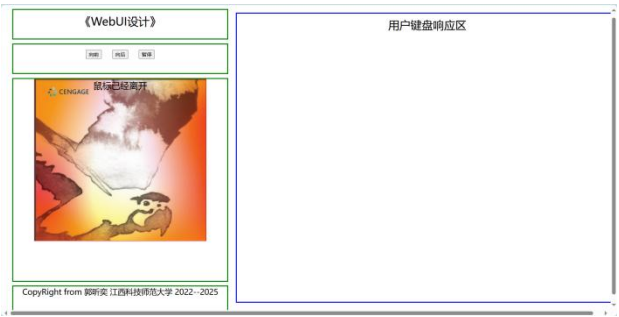


图 5-1 电脑端运行效果图

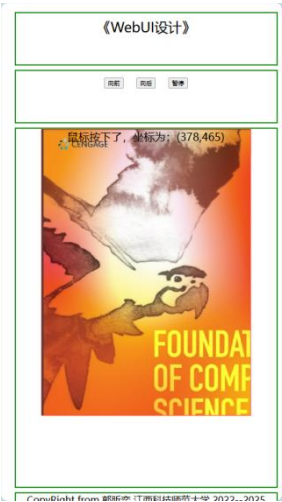
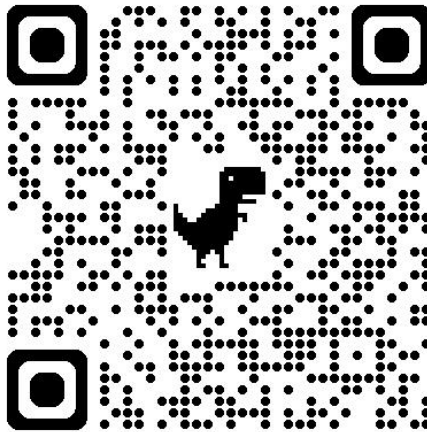
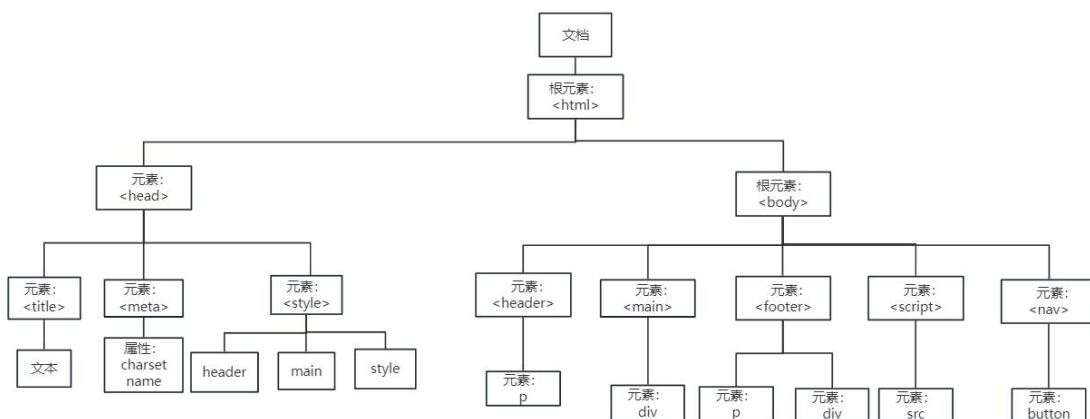


图 5-2 手机端运行效果图

项目地址二维码如下图所示：



项目 dom 树如下图所示：



5.4 项目的代码提交和版本管理

```
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 - baseFont + "px";  
document.body.style.height = UI.appHeight - baseFont*4 + "px";  
if(window.innerWidth<1000){
```

```

    $("#aid").style.display='none';
}
$("#aid").style.width=window.innerWidth-UI.appWidth - baseFont*3 +'px';
$("#aid").style.height= UI.appHeight - baseFont*3 +'px';
//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' ;  
  }  
});
```

接着，我使用 gitbash 工具将以上项目提交，进行第三次迭代

```
git config user.name 郭昕奕  
git config user.email 2509009907@qq.com  
touch 1.3.html myCss.css  
git add 1.3.html myCss.css  
git commit -m 项目第三版：适应互联网时代的响应式设计
```

最后，输入 git log 查看提交日志

```
qcymyg@DESKTOP-G8R5382 MINGW64 /webUI (master)  
$ git log  
commit 45226bcb4001642db2e0af0603537aa1c05b7a5a (HEAD -> master)  
Author: 郭昕奕 <2509009907@qq.com>  
Date:   Wed Jun 12 16:07:52 2024 +0800  
  
    项目第三版：适应互联网时代的响应式设计  
  
commit 16e1d652e29a23a4086a7a06be64b8b0f6e488ef  
Author: 郭昕奕 <2509009907@qq.com>  
Date:   Wed Jun 12 14:49:58 2024 +0800  
  
    项目第二版：移动互联网时代的响应式设计和窄屏实现  
  
commit 6fff01b67db57948f6f0af7071a5a1edee191843  
Author: 郭昕奕 <2509009907@qq.com>  
Date:   Wed Jun 12 08:52:08 2024 +0800  
  
    项目第一版：“三段论”式论文内容设计概要开发
```

第 6 章 个性化 UI 设计中鼠标模型

6.1 分析和设计

为了进一步理解键盘的底层原理，以及实现通过键盘输入从而呈现页面信息，我们搭建了一个用户键盘相应区域进行探索。在 UI 中尝试对鼠标设计控制，设计模拟手机端的触屏效果，设置触屏条件横向移动 100px 为有效拖动，否则为无效拖动。使用鼠标按下、松开、移动模拟手指横向滑动屏幕的操作，添加 Eventlistener 实现 mouseup、mousedown、mouseout 以及 mousemove 的功能。鼠标移动时会显示正在拖动鼠标和拖动距离，当鼠标横向拖动距离大于 100px 显示“鼠标松开！这次是有效拖动！”，拖动距离小于 100px 显示“鼠标松开！”。

这次是无效拖动！”。

6.2 项目的实现和编程

为了页面的直观度，我将用户键盘响应区主要分为了两个部分，中间部分输入字符的内容，尾部用于显示输入字符的 key 值和 keyCode 值。首页，我增文本输入区域，以实现用户键盘的个性化响应。这个任务分为三个部分：首先个文本输入区域，让用户可以在网页中输入文字；其次，通过监听键盘的 keyup 事件，实现输入的即时显示，用户按下键盘时在输入区域中显示相应字母。

```
$("#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' ;
    }
});
```

6.3 项目的运行和测试



图 6-1 电脑运行效果图

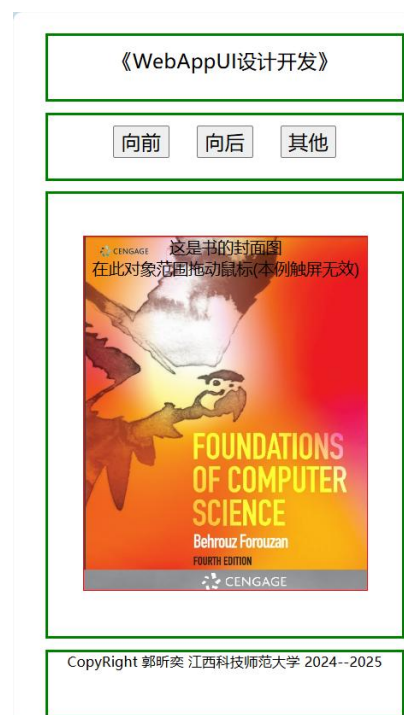
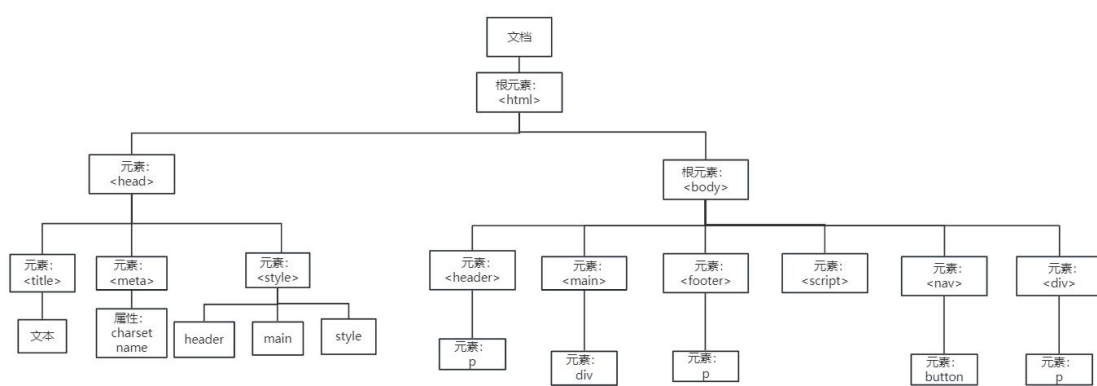


图 6-2 手机端示例图

项目地址二维码如下图所示：



项目 dom 树如下图所示：



6.4 项目的代码提交和版本管理

接着，我使用 gitbash 工具将以上项目提交，进行第三次迭代

```
git config user.name 郭昕奕
git config user.email 2509009907@qq.com
touch 1.4.html myCss.css
git add 1.4.html myCss.css
git commit -m 项目第四版：个性化 UI 设计中的鼠标模型
```

最后，输入 git log 查看提交日志

```
commit b230ffd038c2563d9c45eef483ad36941cd9eb0a (HEAD -> master)
Author: 郭昕奕 <2509009907@qq.com>
Date: Wed Jun 12 16:19:32 2024 +0800
```

项目第四版：个性化UI设计中的鼠标模型

```
commit 45226bcb4001642db2e0af0603537aa1c05b7a5a
Author: 郭昕奕 <2509009907@qq.com>
Date: Wed Jun 12 16:07:52 2024 +0800
```

项目第三版：适应互联网时代的响应式设计

```
commit 16e1d652e29a23a4086a7a06be64b8b0f6e488ef
Author: 郭昕奕 <2509009907@qq.com>
Date: Wed Jun 12 14:49:58 2024 +0800
```

项目第二版：移动互联网时代的响应式设计和窄屏实现

```
commit 6fff01b67db57948f6f0af7071a5a1edee191843
Author: 郭昕奕 <2509009907@qq.com>
Date: Wed Jun 12 08:52:08 2024 +0800
```

项目第一版：“三段论”式论文内容设计概要开发

:

第 7 章 通用的 UI 设计

7.1 分析与设计

在日常的人机交互中，键盘作为输入设备发挥着重要作用。键盘与计算机进行交互的底层逻辑涉及到输入事件的处理和文本传输。当用户按下键盘上的键时，键盘会发送相应的信号给计算机。计算机通过键盘控制器接收这些信号，并将它们转换为数字代码或字符编码，然后发送给操作系统。操作系统接收到键盘输入后，会根据键盘映射表将接收到的代码转换为相应的字符或功能。然后，操作系统将这些字符或功能发送到当前活动的应用程序中。应用程序根据接收到的字符或功能进行相应的处理，比如在文本输入框中显示字符，或者执行与功能键相关的操作。在个性化 UI 设计中，鼠标作为用户与计算机交互的重要工具，其作用不可忽视。鼠标是用户与 UI 界面交互的主要媒介之一，通过点击、拖动、滚动等操作，用户可以快速、准确地与 UI 元素进行交互。鼠标的交互方式和效率直接影响到用户的使用体验。当点击鼠标或触屏时显示鼠标在方框内的具体坐标，并在鼠标或触屏拖动是显示拖动是否有效，当拖动有效时显示鼠标或触屏的拖动到拖动距。

7.2 编程与实现

为了页面的直观度，我将用户键盘响应区主要分为了两个部分，中间部分输入字符的内容，尾部用于显示输入字符的 key 值和 keyCode 值。首页，我增文本输入区域，以实现用户键盘的个性化响应。这个任务分为三个部分：首先个文本输入区域，让用户可以在网页中输入文字；其次，通过监听键盘的 kekeyup 事件，实现输入的即时显示，用户按下键盘时在输入区域中显示相应字段。

```
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 ;
});
}

```

7.3 测试与运行



图 7-1 电脑端运行结果

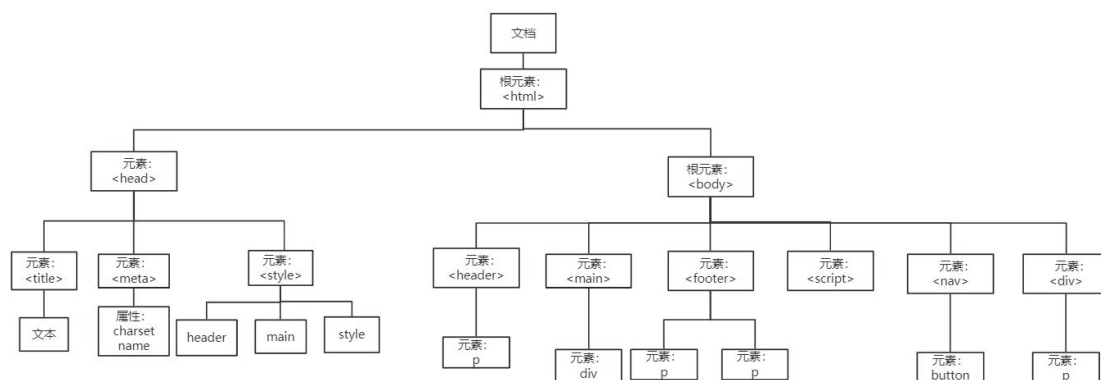


图 7-2 手机端运行实例结果

项目地址二维码如下图所示：



项目 dom 树如下图所示：



7.4 代码提交

接着，我使用 gitbash 工具将以上项目提交，进行第三次迭代

```
git config user.name 郭昕奕
git config user.email 2509009907@qq.com
touch 1.5.html myCss.css
git add 1.5.html myCss.css
git commit -m 项目第五版: 通用的 UI 设计
```

最后，输入 git log 查看提交日志

```
qcymyg@DESKTOP-G8R5382 MINGW64 /webUI (master)
$ git log
commit 92dfd440880d9ff02ce2eaf04b456a0323ed88a3 (HEAD -> master)
Author: 郭昕奕 <2509009907@qq.com>
Date: Wed Jun 12 16:21:00 2024 +0800

    项目第五版: 通用的UI设计

commit b230ffd038c2563d9c45eef483ad36941cd9eb0a
Author: 郭昕奕 <2509009907@qq.com>
Date: Wed Jun 12 16:19:32 2024 +0800

    项目第四版: 个性化UI设计中的鼠标模型

commit 45226bcb4001642db2e0af0603537aa1c05b7a5a
Author: 郭昕奕 <2509009907@qq.com>
Date: Wed Jun 12 16:07:52 2024 +0800

    项目第三版: 适应互联网时代的响应式设计
```

第 8 章 UI 的个性化键盘控制

8.1 分析与设计

UI 的个性化键盘控制的分析与设计是一个关注用户体验、交互设计和技术实现的过程。通过深入了解用户需求、评估技术可行性、分析交互设计等因素，可以设计出符合用户期望的个性化键盘控制方案。在设计过程中，应注重键盘布局、按键功能定制、打字声音和振动反馈等方面的细节优化，以及跨平台兼容性和用户体验的持续提升。我们已经实现了通过个性化 U 对键盘的控制，在本节我们要讨论一下如何通过对鼠标控制来实现页面信息的切换。目前，我们的网页已经可以呈现完整的页面信息了，只不过它的交互性不是很强，而且只能呈现当前屏幕所能显示的信息，显得过于单调。为了丰富网页的信息以及网站本身的互动性，添加翻页功能是十分必要的。

为了进一步理解键盘的底层原理，以及实现通过键盘输入从而呈现页面信息，我们搭建了一个用户键盘相应区域进行探索，如图 8-1 所示。

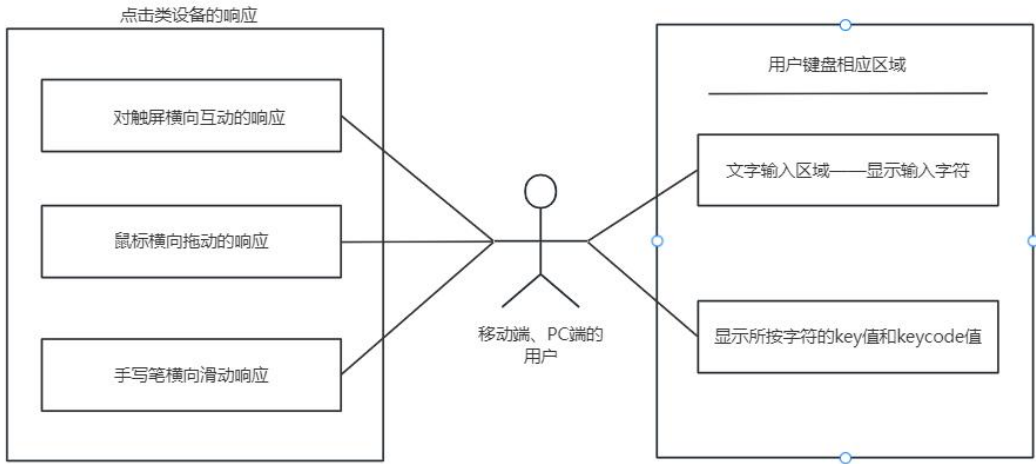


图 8-1 增加个性化键盘响应的用例图

8.2 编程与实现

因为系统中只有一个键盘，所以我们在部署代码时，把键盘事件的监听设置在 DOM 文档最大的可视对象——body 上，通过测试，不宜把键盘事件注册在 body 内部的子对象中。代码如下所示：

```
$( "body" ).addEventListener( "keydown", function( ev ){
    ev.preventDefault() ; //增加“阻止事件对象的默认事件后”，不仅 keypress 事件将不再响应，而且系统的热键，如“F5 刷新页面/Ctrl+R ”、“F12 打开开发者面板”等也不再被响应
    let k = ev.key;
```

```

    let c = ev.keyCode;
    $("#keyStatus").textContent = "按下键 : " + k + " , "+ "编码 : " + c;
  });
  $("body").addEventListener("keyup",function(ev){
    ev.preventDefault() ;
    let key = ev.key;
    $("#keyStatus").textContent = key + " 键已弹起" ;
    if (printLetter(key)){
      $("#typeText").textContent += key ;
    }
    function printLetter(k){
      if (k.length > 1){ //学生须研究这个逻辑的作用
        return false ;
      }
      let puncs =
['~','`','!','@','#','$','%','^','&','*','(',')','-','_','+','=',' ','.',
',',';',':','<','>','?','/',' ','\','\"'] ;
      if ( (k >= 'a' && k <= 'z') || (k >= 'A' && k <= 'Z')
|| (k >= '0' && k <= '9')) {
        console.log("letters") ;
        return true ;
      }
      for (let p of puncs ){
        if (p === k) {
          console.log("puncs") ;
          return true ;
        }
      }
      return false ;
      //提出更高阶的问题，如何处理连续空格和制表键 tab?
    } //function printLetter(k)
  });

```

8.3 测试与运行



图 8-2 电脑端运行效果图

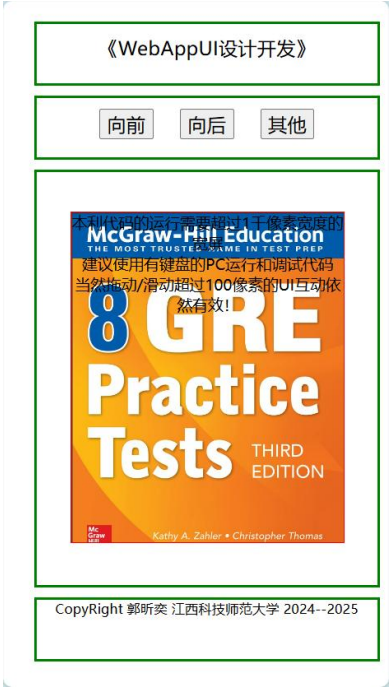
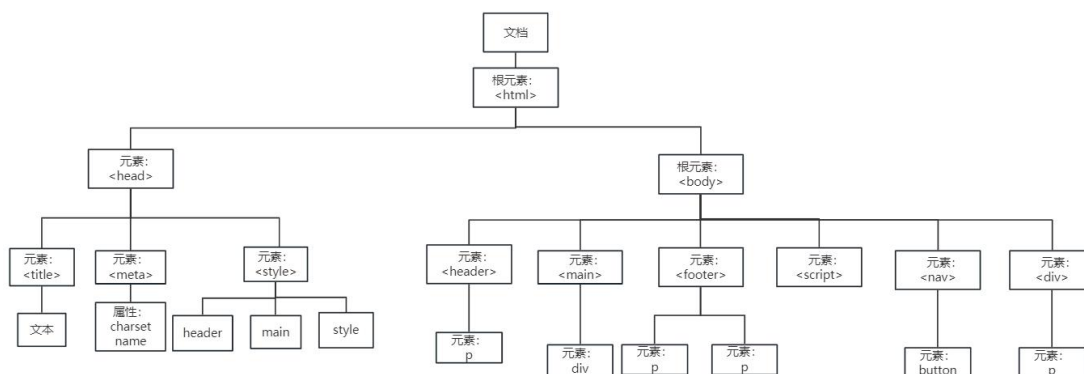


图 8-3 手机端运行效果图

项目地址二维码如下图：



本项目 dom 树如下图所示：



8.3 代码提交

接着，我使用 gitbash 工具将以上项目提交，进行第三次迭代

```
git config user.name 郭昕奕
git config user.email 2509009907@qq.com
touch 1.6.html myCss.css
git add 1.6.html myCss.css
git commit -m 项目第六版: UI 的个性化键盘控制
```

最后，输入 git log 查看提交日志

```
commit 1022d2cfbd7b714a8821551afeb763f7b6784c9c (HEAD -> master)
Author: 郭昕奕 <2509009907@qq.com>
Date:   Wed Jun 12 16:22:34 2024 +0800

    UI中的个性化键盘控制

commit 92dfd440880d9ff02ce2eaf04b456a0323ed88a3
Author: 郭昕奕 <2509009907@qq.com>
Date:   Wed Jun 12 16:21:00 2024 +0800

    项目第五版: 通用的UI设计

commit b230ffd038c2563d9c45eef483ad36941cd9eb0a
Author: 郭昕奕 <2509009907@qq.com>
Date:   Wed Jun 12 16:19:32 2024 +0800

    项目第四版: 个性化UI设计中的鼠标模型

commit 45226bcb4001642db2e0af0603537aa1c05b7a5a
Author: 郭昕奕 <2509009907@qq.com>
Date:   Wed Jun 12 16:07:52 2024 +0800

    项目第三版: 适应互联网时代的响应式设计
```

第 9 章 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^[6]。

下面展示本项目中的高质量代码：

```
var UI = {};  
if(window.innerWidth>600){  
    UI.appWidth=600;  
    }else{  
    UI.appWidth = window.innerWidth;  
    }  
let baseFont: number = UI.appHeight / 20;  
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 - baseFont + "px";  
document.body.style.height = UI.appHeight - baseFont*5 + "px";  
if(window.innerWidth<1000){  
    $("aid").style.display='none';  
}  
$("aid").style.width=window.innerWidth-UI.appWidth - baseFont*3 + 'px';  
$("aid").style.height= UI.appHeight - baseFont*3 + 'px';
```

图 9-1 代码块 1

在以上的代码中，我们创建了一个 UI 对象，用于存储应用程序的宽度和高度，通过改变 body 对象的字体大小来影响其后代，将 body 的高度设置为设备屏幕的高度，从而实现纵向的全屏，在通过 css 对子对象百分比的配合，达到响应式设计的目标，而这样基于屏幕宽度动态设置字体大小可以帮助在不同的设备上保持一致的阅读体验。

```

ev.preventDefault() ;
let key = ev.key;
$("#keyStatus").textContent = key + " 键已弹起" ;
if (printLetter(key)){
  text.textContent += key ;
}
function printLetter(k){
if (k.length > 1){ //学生须研究这个逻辑的作用
  return false ;
}
let puncs = ['~','`','!','@','#','$','%','^','&','*','(',')','_','-','+','=','>','<','>','?','/',' ','\'','\"','\\'];
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 ;

```

图 9-2 代码块 2

以上代码中，定义了一个处理键盘按键事件的功能，key 变量用来存储触发事件的键盘按键值，代码假设在一个键盘案件弹起的事件处理器中运行，并调用 printLetter (key) 函数来检查是否将该字符添加到 text 元素的文本内容中，如果传入字符 k 的长度大于 1，则不打印该字符并返回 false；该代码的优点在于如果需要添加新的字符或者条件，只需要修改 printLetter 函数即可，无需更改其他部分的代码。

第 10 章 用 gitBash 工具管理本项目的 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，它接受键盘输入的命令，并将其传递给操作系统执行。几乎所有 Linux 发行版都提供一个来自 GNU 项目的 shell 程序，名为 Bash。这个名字是 Bourne-Again •Shell 的首字母缩略词，指的是 Bashsh •Shell 的增强替代品，Shell 是 Steve •Boume 编写的原始 Unix •Shell 程序[6]。

Windows 类似，像 Linux 这样的类 Unix 操作系统使用所谓的分层目录结构来组织文件。这意味着文件被组织为树状的目录模式(在其他系统有时称为文件夹)，其中可能包含文件和其他目录。文件系统中的一个目录称为根目录。根目录包含文件和子目录，子目录包含更多的文件和子目录，以此类推。

这种分层目录结构使得 Linux 操作系统的文件管理更加灵活和有条理。用

户可以在命令行中使用 Bash 等 Shell 程序来浏览和操作这些目录和文件，从而实现更加精细的文件管理和系统控制。

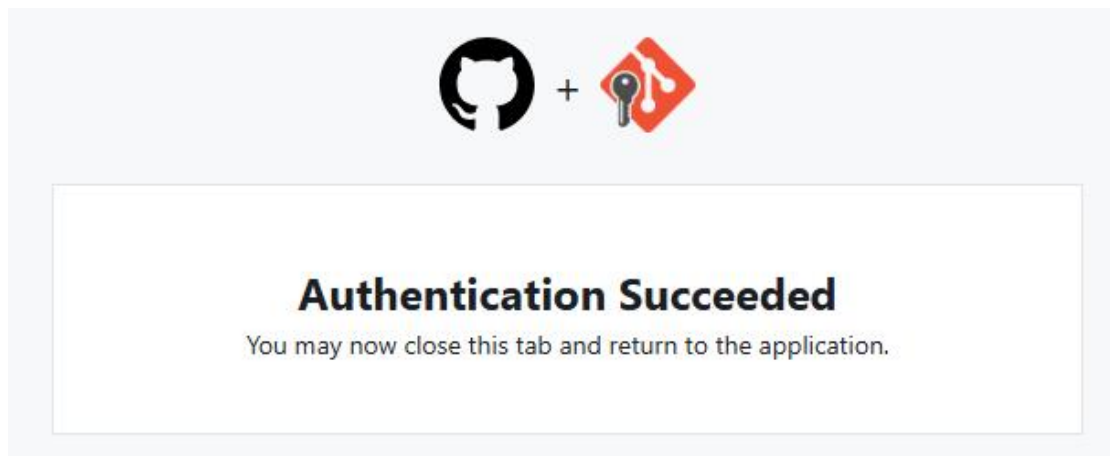
10.2 设置本地仓库和远程代码仓库的链接

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/qcymyg/Ly1in.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：

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。

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