

成绩:

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）：Personalized UI Design and Implementation 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：____蔡欣彤____

学 号：____20193016____

指导教师：____李健宏____

2024 年 6 月 18 日

目录

1. 引言	1
1.1 项目概述	1
1.2 研学计划	1
1.3 研学方法	2
1.3.1 文献综述法	2
1.3.2 原型设计法	2
1.3.3 增量开发法	3
1.3.4 交互设计	3
1.3.5 测试评估	3
1.3.6 版本控制法	4
2. Web 平台和客户端技术概述	4
2.1 Web 平台	4
2.2 Web 编程	4
2.2.1 客户端网页应用的生命周期	5
2.3 项目的增量式迭代开发模式	5
2.3.1 项目运用的开发模式	5
2.2 瀑布模型	6
2.3 增量模型	7
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	8
3.2.1 HTML 代码	8
3.2.2 CSS 代码	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
3.4.1 项目代码的提交	10
3.4.2 项目代码的查询	10
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 响应式设计——适应显示硬件	11
4.2 项目的实现和编程	11
4.3 项目的运行和测试	12
4.4 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	13
5.2 项目的运行和测试	14
5.3 项目的代码提交和版本管理	14
6. 个性化 UI 设计中对鼠标交互的设计开发	14
6.1 项目的实现和编程	15
6.2 项目的运行和测试	16
6.3 项目的代码提交和版本管理	16
7. 对触屏和鼠标的通用交互操作的设计开发	16
7.1 项目的实现和编程	17
7.2 项目的运行和测试	17
7.3 项目的代码提交和版本管理	18

- 8. UI 的个性化键盘交互控制的设计开发18
 - 8.1 分析和设计18
 - 8.2 项目的实现和编程19
 - 8.3 项目的运行和测试20
 - 8.4 项目的代码提交和版本管理20
- 9. 项目中的高质量代码21
- 10. 用 gitBash 工具管理项目的代码仓库和 http 服务器21
 - 10.1 经典 Bash 工具介绍21
 - 10.2 通过 gitHub 平台实现本项目的全球域名21
 - 10.3 创建一个空的远程代码仓库22
 - 10.4 设置本地仓库和远程代码仓库的链接22
- 参考文献24

基于 Web 客户端技术的个性化 UI 的设计和编程

摘要：近十年来，html5 为核心的 web 标准的软件开发技术以其跨平台、开源的优势广泛地运用在各个领域的应用软件开发中。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）。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）。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。

关键词：Web 客户端技术；html5；github；css；javascript；代码编译器 gui；

1. 引言

在信息化快速发展的今天，Web 应用已成为日常生活和工作的核心，Web 应用作为互联网的重要组成部分，其界面设计直接影响用户的使用体验和满意度。传统的 Web 界面设计往往采用统一的风格和布局，缺乏个性化和用户交互的考虑，导致用户在使用过程中感到单调乏味，难以获得良好的体验。因此，如何设计和实现个性化的 UI 界面，提高用户的使用体验和满意度，成为当前 Web 应用设计领域的重要研究方向。

1.1 项目概述

本次课程设计的目标是基于 HTML5 技术，设计和实现一个个性化的用户界面（UI）应用程序。通过运用 HTML5、CSS3 和 JavaScript 等前端技术，我们构建了一个功能丰富、交互流畅的 Web 应用，允许用户根据个人喜好自定义界面风格、布局和交互效果。在项目开发过程中，我们充分利用了 GitHub 这一强大的协作工具，进行代码托管、版本控制、团队协作和代码审查，确保项目开发的高效与规范。我们致力于为用户提供更加符合个人喜好的网页界面，提升整体用户体验。

1.2 研学计划

第 1 周	了解项目要求，分析需求和功能，初步构建应用程序的结构和逻辑。熟悉常用的 UI 设计工具的基本操作，包括画布设置、图层操作、样式设置等。了解响应式设计的重要性，学习如何为不同设备和屏幕尺寸设计 UI。
第 2 周	实现网页窄屏代码和鼠标模型 1.0。显示鼠标的实时坐标。学习如何合理排版页面中的品牌标识、图片、文字等元素，确保用户能够轻松获取信息。
第 3 周	实现鼠标模型 2.0，支持触屏功能。实现页面元素的拖动，显示拖动距离。学习如何构建清晰、易于理解的信息架构，使用户能够快速找到所需信息。设计简单明了的导航菜单，考虑用户的行为习惯，创建合理的层级结构。
第 4 周	实现键盘控制 1.0，显示按下键的字符以及 key 和 keyCode 值。
第 5 周	实现键盘控制 2.0，利用 for 循环，显示单个字符。
第 6 周	实现键盘控制 3.0，实现文本删除与换行功能。
第 7 周	测试 web 应用程序，优化项目代码，全部上传到 GitHub 仓库。

1.3 研学方法

1.3.1 文献综述法

文献综述法是我们进行研学活动的起点和基础。在此阶段，我们将广泛搜集、整理和分析关于 Web 客户端技术、个性化 UI 设计以及实现方法的相关文献。这些文献可以来自学术期刊、技术博客、专业书籍、在线论坛等。通过对这些文献的深入研究，我们可以全面了解 Web 客户端技术的最新发展趋势、个性化 UI 设计的原则和最佳实践，以及在实际开发中常用的技术和工具。同时，文献综述还能帮助我们识别出当前领域的研究热点和难点，为后续的研究工作提供方向。

1.3.2 原型设计法

原型设计法是一种在 UI 设计过程中常用的方法。在个性化 UI 设计中，我们将采用原型设计法来快速构建出 UI 设计的初步方案。通过原型设计，我们可以将抽象的设计概念转化为具体的视觉形式，让开发者和用户能够直观地看到设计效果。同时，原型设计还可以帮助我们进行用户测试和反馈收集，以便及时发现设计中存在的问题并进行改进。在原型设计过程中，我们将使用专业的设计工具

（如 Sketch、Figma 等）来创建高保真度的 UI 设计原型，并通过多次迭代和优化来完善设计方案。

1.3.3 增量开发法

增量开发法是一种将软件开发过程划分为一系列小步骤的方法。在个性化 UI 设计和实现的过程中，我们将采用增量开发法来逐步推进项目的开发进度。具体而言，我们将根据原型设计的结果，将 UI 设计划分为若干个模块或组件，并为每个模块或组件制定详细的开发计划。然后，我们将按照计划逐步开发这些模块或组件，并在每个阶段进行严格的测试和评估。通过增量开发法，我们可以确保项目的质量和进度得到有效控制，同时降低开发风险。

1.3.4 交互设计

交互设计是个性化 UI 设计中至关重要的一环。在交互设计过程中，我们将关注用户与 Web 应用之间的交互方式和流程。我们将深入研究用户的操作习惯、心理需求和期望，并据此设计出符合用户需求的交互方式和流程。具体而言，我们将关注以下几个方面：

- ①导航设计：设计清晰、直观的导航结构，帮助用户快速找到所需信息。
- ②信息架构：合理组织页面内容，确保信息的层次结构和逻辑关系清晰明了。
- ③交互反馈：及时响应用户操作，并给予明确的反馈，提高用户的使用体验。
- ④可用性测试：通过用户测试来评估交互设计的有效性，并根据测试结果进行改进。

1.3.5 测试评估

测试评估是确保 Web 应用质量和用户体验的关键环节。在个性化 UI 设计和实现的过程中，我们将采用多种测试方法来评估 Web 应用的质量和性能。具体而言，我们将进行以下测试：

- ①功能测试：验证 Web 应用的功能是否按照需求规格说明书正确实现。
- ②性能测试：测试 Web 应用的响应时间、吞吐量、并发用户数等性能指标，确保应用能够满足用户需求。
- ③兼容性测试：测试 Web 应用在不同浏览器、操作系统和设备上的兼容性。
- ④安全性测试：测试 Web 应用的安全性，包括防止 SQL 注入、跨站脚本攻击等安全漏洞。

⑤用户体验测试：通过用户测试来评估 Web 应用的易用性、美观度和用户满意度等指标。

1.3.6 版本控制法

版本控制法是一种用于管理代码变更的分布式版本控制系统。在个性化 UI 设计和实现的过程中，我们将采用版本控制法来管理项目的代码和文档。我们将为每个项目创建一个 Git 仓库，并在开发过程中将代码和文档提交到仓库中进行版本控制。通过 G 版本控制法，我们可以方便地追踪代码的变更历史、协作开发、合并代码和回滚错误等。同时，Git 还支持分支管理功能，允许我们在不影响主分支的情况下进行新功能或修复错误的开发。这有助于我们保持项目的稳定性和可维护性。

2. Web 平台和客户端技术概述

2.1 Web 平台

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，几乎完美缔造了全球开发者实现开发平台统一的理想，直到今天，科学家与 Web 行业也还一直在致力于完善这个伟大而光荣的理想[1]。学习 Web 标准和 Web 技术，学习编写 Web 程序和应用有关工具，最终架构一套高质量代码的跨平台运行的应用，是我的毕设项目应用的技术路线。

Web 平台，又称为全球广域网或万维网（World Wide Web），是一个庞大的、分布式的、基于超文本的信息系统。该平台由各种网站、网页和应用构成，为用户提供丰富的信息资源和服务。Web 平台是全球性的信息交流平台，允许用户通过浏览器访问各种网络资源。它具有高度的跨平台性，用户可以在不同操作系统和设备上访问 Web 应用。

2.2 Web 编程

Web 编程是一个很大的领域，不同类型的 Web 编程由不同的工具实现。但是，基本上所有的工具都使用核心语言 HTML。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）；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。

2.2.1 客户端网页应用的生命周期

客户端网页应用的生命周期从用户指定网站地址(或点击链接)开始，到用户离开网页为止。它包含两个步骤:页面构建和事件处理。在页面构建过程中，浏览器发送请求到服务器，获取并解析 HTML、CSS 和 JavaScript 文件，构建并渲染页面。在事件处理阶段，用户与页面交互，通过事件触发 JavaScript 代码来动态更新页面内容，如图 2.1 所示。

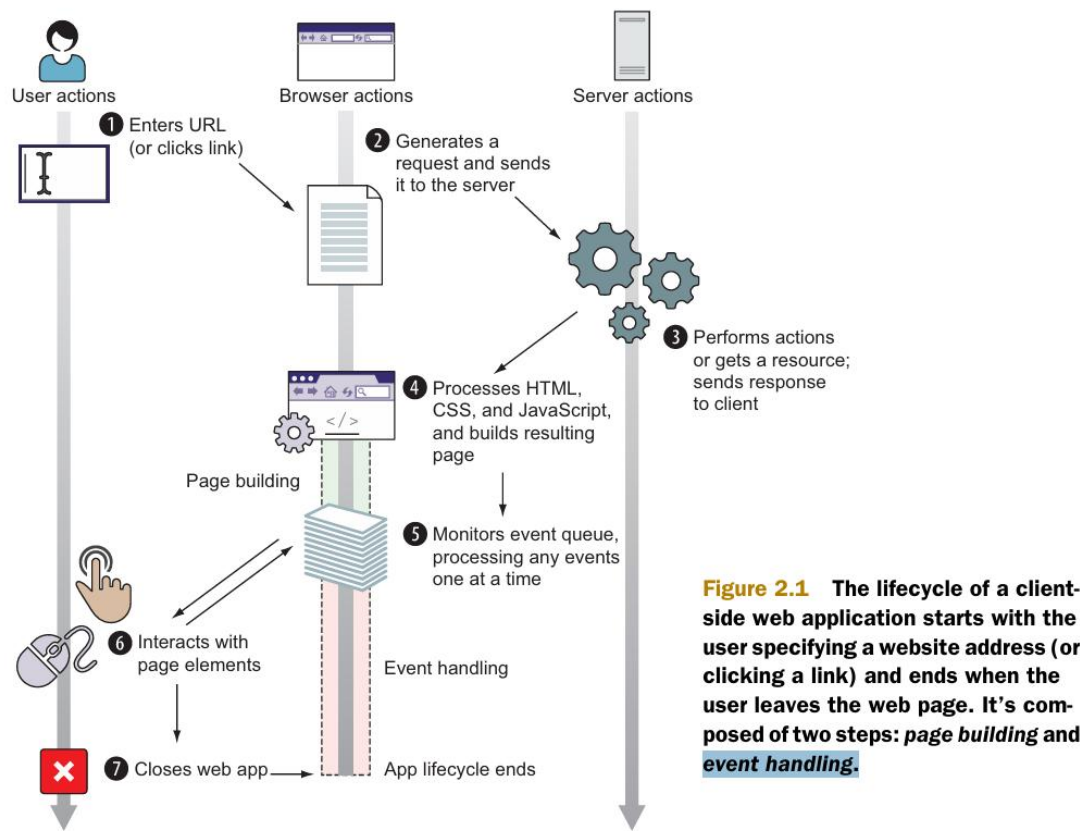


图 2.1 客户端网页应用的生命周期

2.3 项目的增量式迭代开发模式

2.3.1 项目运用的开发模式

从软件工程的管理视角来看待和规范项目的编写过程，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：

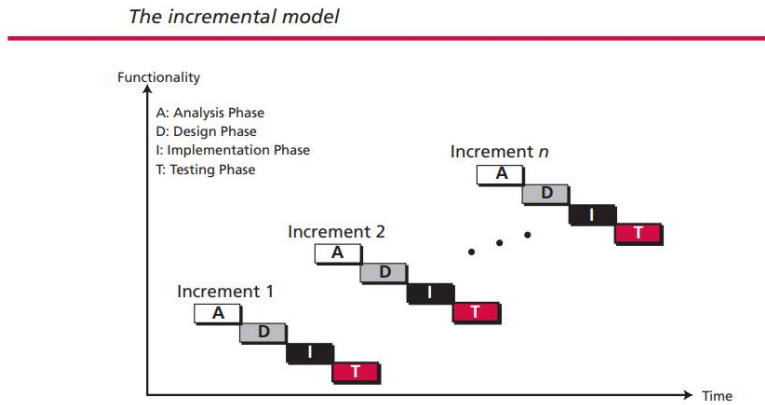


图 2.1 增量模型

2.2 瀑布模型

瀑布模型是 20 世纪 80 年代之前最受推崇的软件开发模型，它是一种线性的开发模型，具有不可回溯性。开发人员必须等前一阶段的任务完成后，才能开始进行后一阶段的工作，并且前一阶段的输出往往就是后一阶段的输入。由于它的不可回溯性，如果在软件生命周期的后期发现并要改正前期的错误，那么需要付出很高的代价。传统的瀑布模型是文档驱动的，如图 2.2 所示。

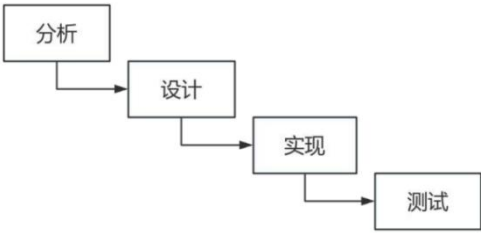


图 2.2 瀑布模型

瀑布模型的优点是过程模型简单，执行容易；缺点是无法适应变更。瀑布模型是一种注重规划和文档化、强调阶段明确和单向转移、严格控制变更和强调质量和测试的软件开发过程模型。它适用于需求稳定、可预测的大型、复杂软件项目，但开发周期较长，无法适应快速变化的需求。

2.3 增量模型

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。

一般情况下，开发人员会首先实现提供基本核心功能的增量组件，创建一个具备基本功能的子系统，然后再对其进行完善。增量模型如图 2.3 所示。

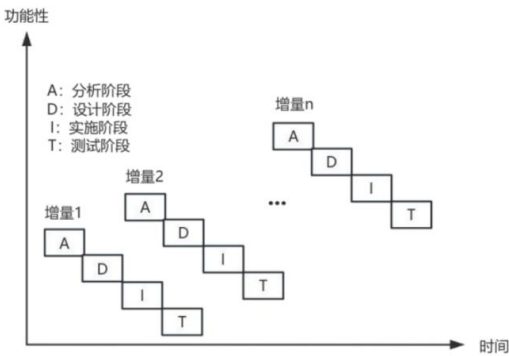


图 2.3 增量模型

增量模型的最大特点就是待开发的软件系统模块化和组件化。

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 3.1 用例图所示：



图 3.1 用例图

3.2 项目的实现和编程

3.2.1 HTML 代码

```
<header>
    《 个性化 UI 设计和实现 》
</header>
<main>
    本次课程设计的目标是基于 HTML5 技术，
    设计和实现一个个性化的用户界面（UI）应用程序。
    通过运用 HTML5、CSS3 和 JavaScript 等前端技术，
    构建一个功能丰富、交互流畅的 Web 应用，
    允许用户根据个人喜好自定义界面风格、布局和交互效果。
    在项目开发过程中，我们充分利用了 GitHub 这一强大的协作工具，
    进行代码托管、版本控制、团队协作和代码审查，确保项目开发的高效与规范。
    我们致力于为用户提供更加符合个人喜好的网页界面，提升整体用户体验。
</main>
<footer>
    Code by 蔡欣彤 江西科技师范大学 2024--2025
</footer>
```

图 3.2 HTML 代码块

3.2.2 CSS 代码

```
<style>
*{
    margin: 10px;
    text-align: center;
    font-size:30px ;
}
header{
    border: 2px solid blue;
    height: 150px;
    font-size: 1.5em;
}
```

```
main{
  border: 2px solid blue;
  height: 550px;
}
footer{
  border: 2px solid blue;
  height: 50px;
}

a{
  display: inline-block ;
  padding:10px ;
  color: white;
  background-color: blue;
  text-decoration: none ;
}
</style>
```

图 3.3 CSS 代码块

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 3.4 所示。由于本项目的阶段性文件已经上传 github 网站，移动端用户可以通过扫描图 3.5 的二维码，运行测试本项目的第一次开发的阶段性效果。

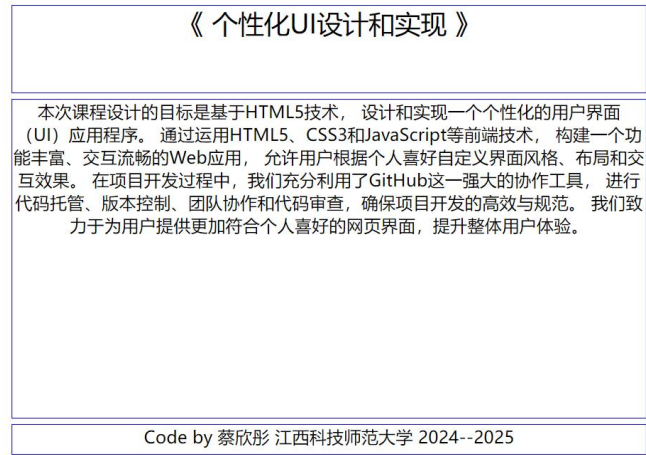


图 3.4 PC 端运行效果图

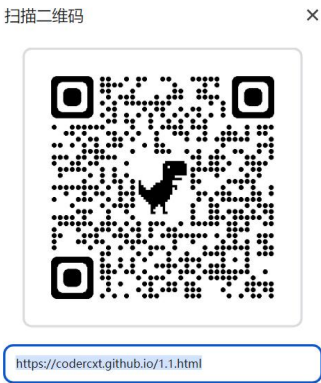


图 3.5 移动端二维码

3.4 项目的代码提交和版本管理

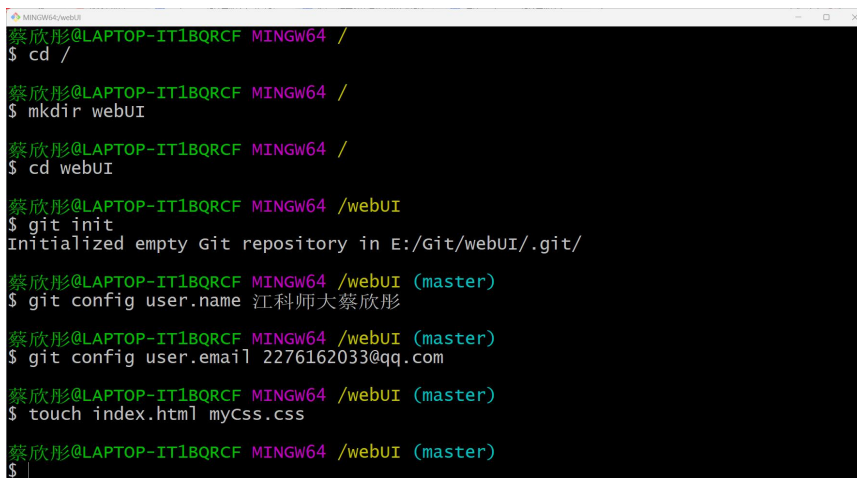
本项目的文件通过 gitBash 工具管理，作为项目的第一次迭代，在代码提交和版本管理环节，我们的目标是建立项目的基本文件结构，还有设置好代码仓库

的基本信息：如开发者的名字和电子邮件。

3.4.1 项目代码的提交

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 江科师大蔡欣彤  
$ git config user.email 2276162033@qq.com  
$ touch index.html myCss.css
```

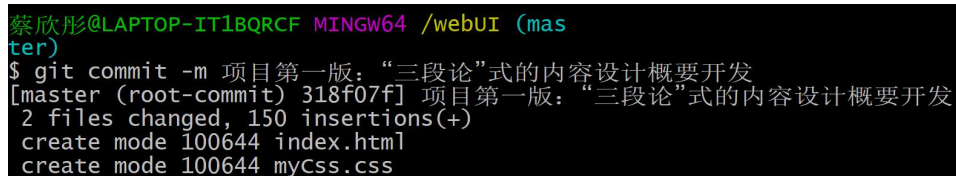


```
蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /  
$ cd /  
蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /  
$ mkdir webUI  
蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /  
$ cd webUI  
蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI  
$ git init  
Initialized empty Git repository in E:/Git/webUI/.git/  
蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (master)  
$ git config user.name 江科师大蔡欣彤  
蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (master)  
$ git config user.email 2276162033@qq.com  
蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (master)  
$ touch index.html myCss.css  
蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (master)  
$
```

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：



```
蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (master)  
$ git commit -m 项目第一版：“三段论”式的内容设计概要开发  
[master (root-commit) 318f07f] 项目第一版：“三段论”式的内容设计概要开发  
2 files changed, 150 insertions(+)  
create mode 100644 index.html  
create mode 100644 myCss.css
```

3.4.2 项目代码的查询

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：

```
蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (master)
$ git log
commit 318f07ffa4396798a397a3ad663e4d065eabdebb (HEAD -> master)
Author: 江科师大蔡欣彤 <2276162033@qq.com>
Date: Fri Jun 21 17:36:11 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

4.1 响应式设计——适应显示硬件

计算机所使用的显示硬件千差万别，显示器的大小和分辨率取决于成本。设计师们选择让网页给出一般的布局准则，并允许浏览器选择如何在给定的计算机上显示页面，而不是为每种类型的显示提供每个网页的版本。因此，一个网页不能提供很多细节。例如，网页的作者可以指定一组句子组成一个段落，但作者不能指定诸如一行的确切长度或是否缩进段落开头等细节。[1]

允许浏览器选择显示细节会产生一个有趣的结果：当通过两个浏览器或在硬件不同的两台计算机上浏览时，网页可能会显示不同的内容。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：网页给出了关于期望呈现的一般指导方针；浏览器在显示页面时选择详细信息。因此，同一网页在两台不同的计算机或不同的浏览器上显示时可能会略有不同。[1]

4.2 项目的实现和编程

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。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。如代下图代码块 4.1 所示：

```
<style>
*{
  margin: 10px;
  text-align: center;
}
header{
  border: 2px solid black;
  height: 13%;
  font-size: 1.4em;
}
```

```

main{
  border: 2px solid black;
  height: 70%;
  font-size: 1.2em;
}
nav{
  border: 2px solid black;
  height: 10%;
}
nav button{
  font-size: 1.1em;
}
footer{
  border: 2px solid black;
  height: 6%;
}
</style>

```

图 4.1 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。如代下图代码块 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>

```

图 4.2 代码块 4.2

4.3 项目的运行和测试

项目第 2 次开发的运行结果如下图 5.3 所示。移动端用户可以通过扫描图 4.4 的二维码，运行测试本项目的第 2 次开发的阶段性效果。



图 4.3 运行效果图

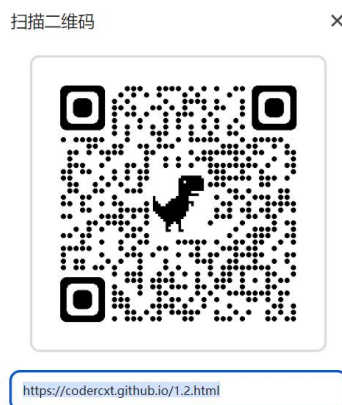


图 4.4 移动端二维码

4.4 项目的代码提交和版本管理

```
蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (master)
$ git log
commit 05e4274b7a52c4ede0472a2a6f65080f8e69a54c (HEAD -> master)
Author: 江科师大蔡欣彤 <2276162033@qq.com>
Date: Sat Jun 22 01:16:00 2024 +0800

项目第二版：移动互联时代的响应式设计（窄屏）
```

5. 应用响应式设计技术开发可适配窄屏和宽屏的 UI

适用移动互联时代阐述移动互联时代的用户终端的多样性，以及如何用 css 语言和 JavaScript 语言实现响应式设计

5.1 项目的实现和编程

```
//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;
    $("keyboard").textContent = "您的按键 : " + k + " , "+" 字符编码 : " + c;
});

```

图 5.1 代码块 5.1

5.2 项目的运行和测试

项目第3次开发的运行结果如下图5.2所示。移动端用户可以通过扫描图5.3的二维码，运行测试本项目的第3次开发的阶段性效果。

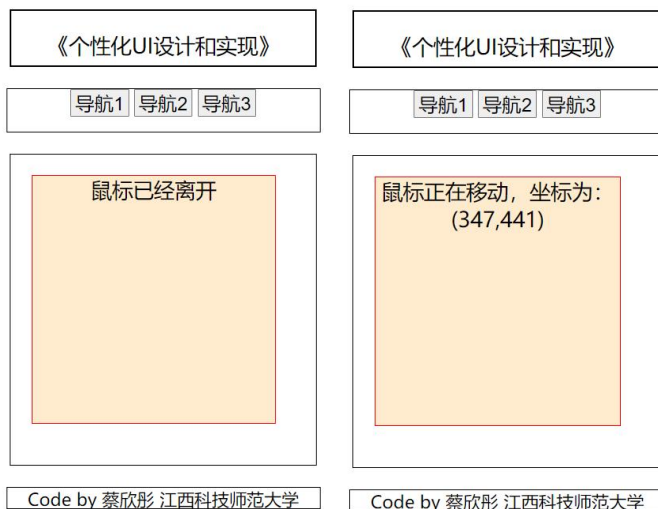


图 5.2 运行效果图

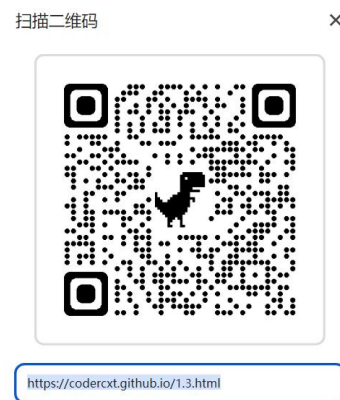


图 5.3 移动端二维码

5.3 项目的代码提交和版本管理

```

蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (master)
$ git log
commit d2b3fc1f15b46ddd94f909f401d7626a408fcb72 (HEAD -> master)
Author: 江科师大蔡欣彤 <2276162033@qq.com>
Date: Sat Jun 22 01:26:34 2024 +0800

```

项目第三版：应用响应式设计技术开发可适配窄屏和宽屏的UI，显示鼠标坐标

6. 个性化 UI 设计中对鼠标交互的设计开发

6.1 项目的实现和编程

```
//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' ;
  }
});

```

图 6.1 代码块 6.1

6.2 项目的运行和测试

项目第 4 次开发的运行结果如下图 6.2 所示。移动端用户可以通过扫描图 6.3 的二维码，运行测试本项目的第 4 次开发的阶段性效果。

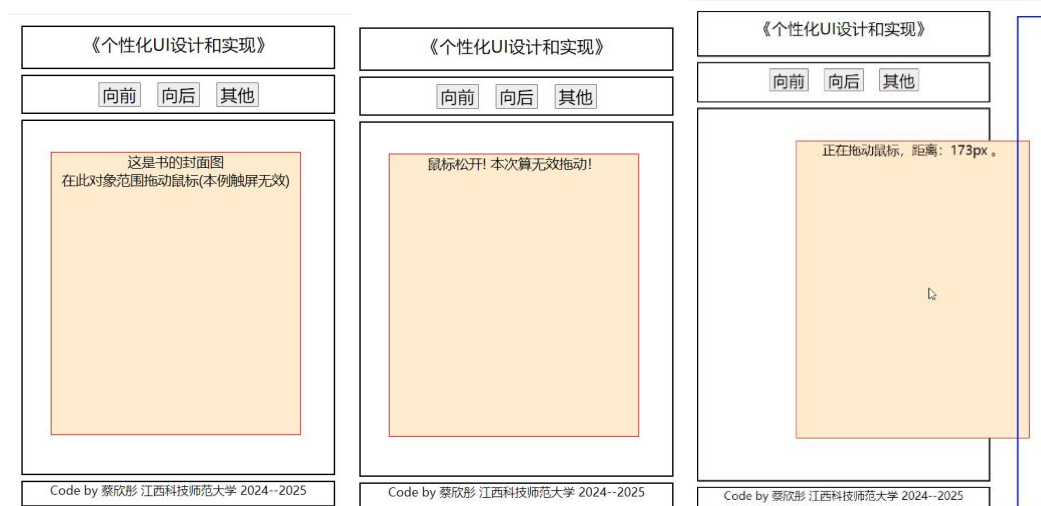


图 6.2 运行效果图

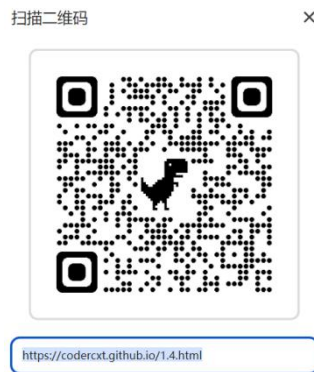


图 6.3 移动端二维码

6.3 项目的代码提交和版本管理

```

蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (master)
$ git log
commit afb2092f326d71a0760556ea3c21a5b970ff5316 (HEAD -> master)
Author: 江科师大蔡欣彤 <2276162033@qq.com>
Date: Sat Jun 22 01:31:50 2024 +0800

    项目第四版：个性化UI设计中对鼠标交互的设计开发，让鼠标可以拖动页面

```

7. 对触屏和鼠标的通用交互操作的设计开发

7.1 项目的实现和编程

```
$("#body").addEventListener("keypress", function(ev){
    $("#firstLine").textContent += ev.key ;
});
$("#body").addEventListener("keydown", function (ev) {
    let k = ev.key;
    let c = ev.keyCode;
    $("#status").textContent = "您已按下键 : " + k + " , " + "字符编码 : " + c;
});
```

图 7.1 代码块 7.1 字符的显示

```
/*判断是否是单个字符*/
function printLetter(k) {
    //判断字符串长度是否大于 1
    if (k.length > 1) {
        return false;
    }
    let puncs = ['~', '`', '!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '-', '_', '+', '=', ',', '.', '<', '>', '?', '/', '"', ' '];
    /*字母输出*/
    if ((k >= 'a' && k <= 'z') || (k >= 'A' && k <= 'Z') || (k >= '0' && k <= '9')) {
        return true;
    }
    /*符号输出*/
    for (let p of puncs) {
        if (p === k) {
            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?
}
```

图 7.2 代码块 7.2 判断是否为单个字符

7.2 项目的运行和测试

项目第 5 次开发的运行结果如下图 7.3 所示。移动端用户可以通过扫描图 7.4 的二维码，运行测试本项目的第 5 次开发的阶段性效果。

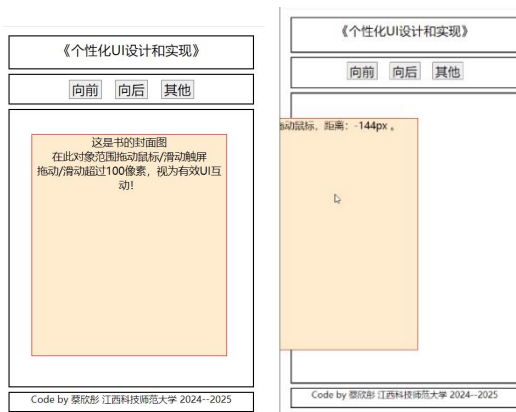


图 7.3 运行效果图

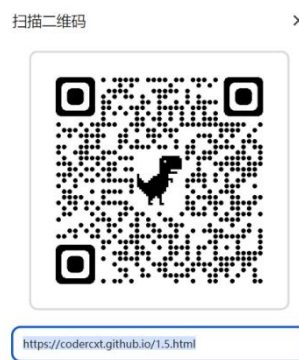


图 7.4 移动端二维码

7.3 项目的代码提交和版本管理

```
蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (master)
$ git log
commit 35d30ab4c575d37e38fea365a9aabb430f93e97b (HEAD -> master)
Author: 江科师大蔡欣彤 <2276162033@qq.com>
Date: Sat Jun 22 01:39:57 2024 +0800

    项目第五版：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

个性化键盘交互控制的设计，旨在为用户提供一种更加便捷、高效且符合个人使用习惯的输入方式。通过深入分析用户的使用习惯和需求，设计出一套具有高度可定制性和易用性的键盘交互方案。

自定义热键，允许用户根据自己的使用习惯和需求，自定义键盘上的热键组合。这种设计方式可以大大提高用户的工作效率，减少不必要的操作步骤。在键盘界面上设置快捷操作栏，用户可以通过简单的点击或滑动操作，快速执行常用的功能或任务。根据用户的输入习惯和手指大小，优化键盘的布局和按键大小。确保用户在输入时能够轻松、准确地按下每个按键。

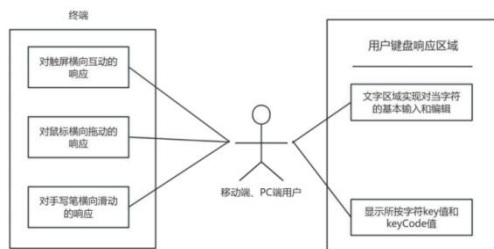


图 8.1 UI 的个性化键盘响应用例图

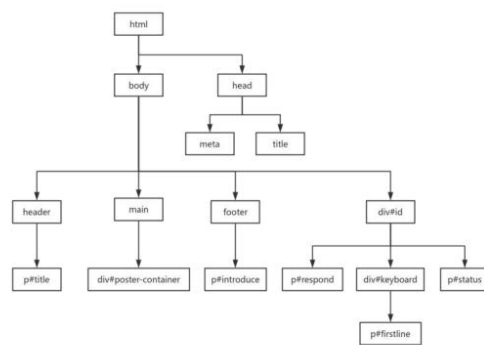


图 8.2 UI 的个性化键盘响应 DOM 树

8.2 项目的实现和编程

明确个性化键盘交互的具体需求，如自定义快捷键、智能预测、自适应布局等。使用 HTML 和 CSS 设计键盘的基本布局和样式。使用 JavaScript（编写代码来处理键盘的交互逻辑。包括监听键盘事件（如 `keydown`、`keyup` 等），处理自定义快捷键，实现智能预测等。在多种浏览器和设备上测试个性化键盘的交互效果，并根据测试结果进行优化。如果设计中包含多模式切换功能，需要编写代码以支持不同模式下的键盘布局和交互逻辑。

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：

```
/*实现输入 Enter 键进行换行， BackSpace 进行删除*/
$("body").addEventListener("keydown", function (ev) {
    let k = ev.key;
    let c = ev.keyCode;
    $("status").textContent = "您已按下键 : " + k + " , " + "字符编码 : "
+ c;
});
$("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;

    /*判断键盘按下的是否为 Enter 键，如果是添 p 元素实现换行*/
    if (k === "Enter") {
        //有且只能在 document 中创建子节点
        let p = document.createElement("p");
        //通过创建 p 元素，添加子节点来实现换行。
        $("keyboard").appendChild(p);
    }
});
```

```

    } else if (k === "Backspace") {
        /*没有字符可以删除了，则将该子节点删除*/
        if ($("#keyboard").lastElementChild.textContent === "") {
            /*删除前保证 keyboard 中至少有一个字节*/
            if ($("#keyboard").childElementCount > 1) {
                $("#keyboard").removeChild($("#keyboard").lastElementC
child);
            }
        } else {
            $("#keyboard").lastElementChild.textContent =
$("#keyboard").lastElementChild.textContent.slice(0, -1);
        }
    } else if (printLetter(k)) {
        $("#keyboard").lastElementChild.textContent += k;
    }
}

```

图 8.3 代码块 8.3 实现换行与删除

8.3 项目的运行和测试

项目第 6 次开发的运行结果如下图 8.4 所示。移动端用户可以通过扫描图 8.5 的二维码，运行测试本项目的第 6 次开发的阶段性效果。



图 8.4 运行效果图

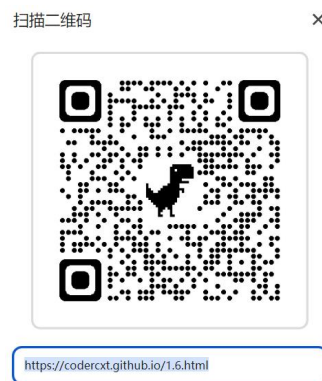


图 8.5 移动端二维码

8.4 项目的代码提交和版本管理

```

蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (master)
$ git log
commit 107ccbb578a77497cb3106c7df4323121761e177 (HEAD -> master)
Author: 江科师大蔡欣彤 <2276162033@qq.com>
Date: Sat Jun 22 01:45:50 2024 +0800

    项目第六版: UI的个性化键盘交互控制的设计开发

```


9. 项目中的高质量代码

如今，电脑就像螺丝刀一样常见，但它们要复杂得多，让它们做你想让它们做的事情并不总是那么容易。如果你给你的电脑的任务是一个常见的，很容易理解的任务，比如显示你的电子邮件或像计算器一样工作，你可以打开适当的应用程序并开始工作。但对于独特的或开放式的任务，可能没有应用程序。

这就是编程可能发挥作用的地方。编程是构建程序的行为——一组告诉计算机该做什么的精确指令。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，甚至享受用愚蠢的机器可以处理的方式思考的严谨性，编程是有回报的。它可以让你在几秒钟内完成手工永远做不完的事情。它是一种让你的计算机工具做它以前不能做的事情的方法。它还提供了一个很好的抽象思维练习。[6]

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 是一个接受键盘命令并将其传递给操作系统执行的程序。几乎所有的 Linux 发行版都提供了一个来自 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 bash 是 sh 的增强替代品，sh 是 Steve Bourne 编写的原始 Unix shell 程序。[7]

与 Windows 一样，像 Linux 这样的类 unix 操作系统用所谓的分层目录结构来组织文件。这意味着它们被组织成树状的目录模式(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，子目录包含更多的文件和子目录，以此类推。[7]

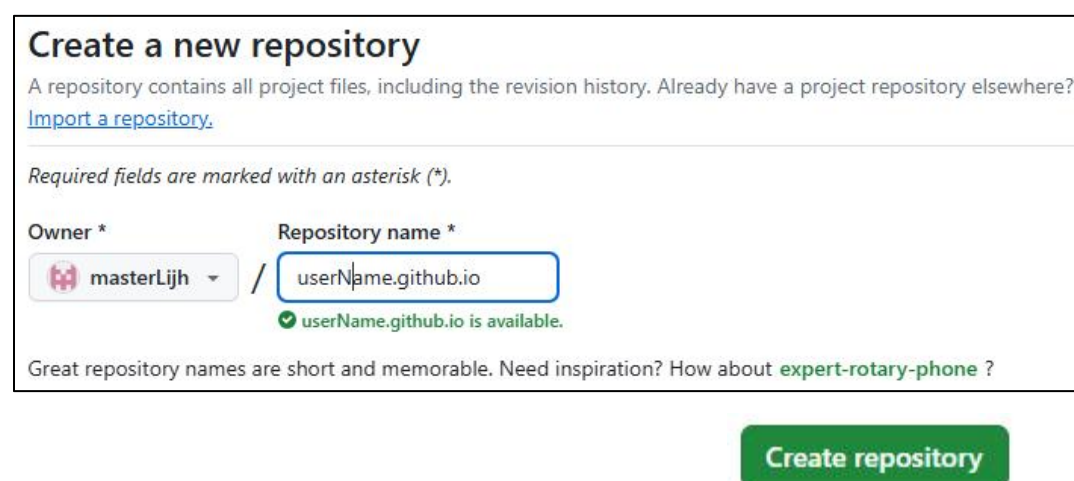
10.2 通过 gitHub 平台实现本项目的全球域名

通过 GitHub 平台实现本项目的全球域名服务，是一项创新且高效的方式，为项目提供了稳定、可靠且全球可达的在线展示和访问途径。GitHub Pages 作为 GitHub 提供的静态站点托管服务，允许用户直接从 GitHub 仓库获取 HTML、CSS 和 JavaScript 文件，并通过简单的配置，即可将项目部署为在线网站。

利用 GitHub Pages，本项目能够拥有一个独特的全球域名，形如 `projectname.github.io`，其中 `projectname` 为项目在 GitHub 上的名称。这一域名不仅易于记忆和分享，而且具有全球可达性，无论用户身处何地，只需在浏览器中输入该域名，即可轻松访问到项目的内容。

此外，GitHub Pages 还提供了自定义域名的功能，用户可以将自己的域名与 GitHub Pages 站点进行绑定，从而进一步提升项目的品牌影响力和专业度。通过这一功能，用户可以将自己的域名解析到 GitHub Pages 提供的 IP 地址上，实现自定义域名的全球访问。

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 * / Repository name *

 masterLijh /

✓ 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：

```
MINGW64/webUI
蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /
$ cd webUI/

蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (master)
$ echo "webUI应用的远程http服务设置">> README.md

蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (master)
$ git init
Reinitialized existing Git repository in E:/Git/webUI/.git/

蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (master)
$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory
```

```

蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (master)
$ git commit -m "这是我第一次把代码仓库上传到github平台"
[master 467ef26] 这是我第一次把代码仓库上传到github平台
1 file changed, 1 insertion(+)
create mode 100644 README.md

蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (master)
$ git branch -M main

蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (main)
$ git remote add origin https://github.com/codercxt/codercxt.github.io

蔡欣彤@LAPTOP-IT1BQRCF MINGW64 /webUI (main)
$ 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 10.1 所示：



图 10.1 开发者初次授权图

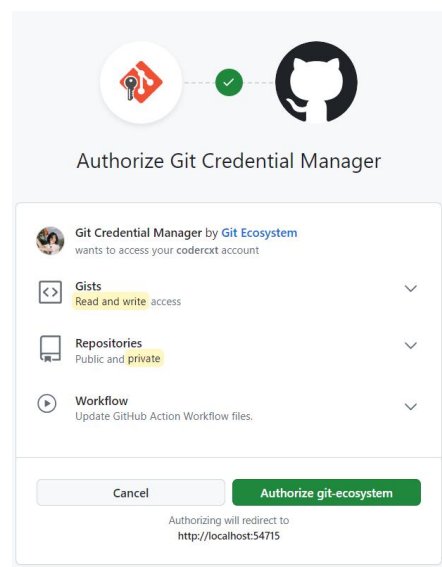
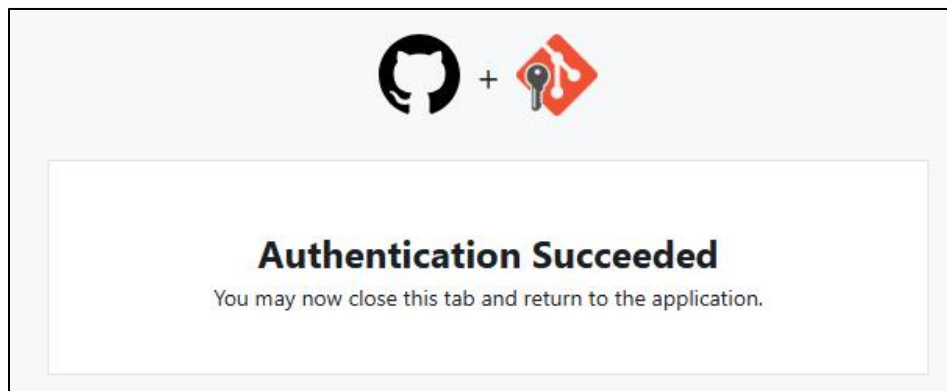


图 10.2 开发者再次授权图

再次确认授权 gitBash 拥有访问改动远程代码的权限，如上图 10.2 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。

参考文献

- [1] W3C.W3C's history.W3C Community.[EB/OL].<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.2019:6
- [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

Personalized UI Design and Implementation Based on Web Client Technology

Abstract:In recent ten years,html5 as the core of the web standard software development technology with its cross-platform,open source advantages are widely used in various fields of application software development.This project chooses the web client technology of html5 as the technical route to carry out the research and practice of program design and software development.Through extensive access to technical books,developer forums, and literature,a personalized user interface (UI) application was designed and developed.In the development,html language was comprehensively applied to conduct content modeling,css language to develop UI appearance design,and javascript language programming to realize UI interaction functions.In addition to directly using the API at the lowest level of the web client,each code of the project was written manually one by one,without importing any code (framework and library) of others.From the perspective of engineering management,the incremental development mode adopted in this project has carried out six incremental code reconstructions (A:Analysis,D:Design,I: Implementation,T:Testing) in a way of gradual refinement.From the perspective of open source and sharing of code,this project adopts git tool for version management,rebuilds the code six times in the development process and formally submits the code,and modifies and submits the code twice in the test.Finally,gitbash tool is used to upload the code warehouse of this project to the famous github.Using the http server provided by github,this project realizes the deployment of UI application on the global Internet,and we can conveniently and efficiently access this program across platforms through the address and two-dimensional code.

Key Words: Web client technology; html5; github; css; javascript;Code compiler gui;