目录

[基于Web客户端技术的个性化UI的设计和实现 2](#_Toc23540)

[（Customized UI design and Programming based on Web client technology） 2](#_Toc12312)

[1. 前言 2](#_Toc20116)

[1.1毕设任务分析 2](#_Toc7639)

[1.2 研学计划 2](#_Toc32150)

[1.3研究方法 3](#_Toc19631)

[2. 技术总结和文献综述 3](#_Toc20372)

[2.1 Web平台和客户端技术概述 3](#_Toc1219)

[2.2 项目的增量式迭代开发模式 4](#_Toc16464)

[3. 内容设计概要 6](#_Toc32336)

[3.1 分析和设计 6](#_Toc11398)

[3.2 项目的实现和编程 6](#_Toc31412)

[3.3 项目的运行和测试 7](#_Toc4460)

[3.4 项目的代码提交和版本管理 8](#_Toc26220)

[4. 移动互联时代的UI开发初步——窄屏终端的响应式设计 9](#_Toc1315)

[5. 应用响应式设计技术开发可适配窄屏和宽屏的UI 11](#_Toc21579)

[6. 个性化UI设计中对鼠标交互的设计开发 11](#_Toc6042)

[7. 对触屏和鼠标的通用交互操作的设计开发 11](#_Toc25410)

[8. UI的个性化键盘交互控制的设计开发 11](#_Toc1982)

[9. 谈谈本项目中的高质量代码 12](#_Toc27488)

[10. 用gitBash工具管理项目的代码仓库和http服务器 13](#_Toc22245)

[10.1 经典Bash工具介绍 13](#_Toc29698)

[10.2 通过gitHub平台实现本项目的全球域名 13](#_Toc21719)

[10.3 创建一个空的远程代码仓库 14](#_Toc17477)

[10.4 设置本地仓库和远程代码仓库的链接 14](#_Toc20710)

[参考文献： 18](#_Toc10291)

[写作指导： 18](#_Toc16622)

# 基于Web客户端技术的个性化UI的设计和实现

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

科师大元宇宙产业学院2021级 XXX

【摘要】

近年来，随着移动互联网时代的发展，HTML5的Web技术在软件开发领域扮演着重要的角色。本文探讨了一项以HTML5的web客户端技术为核心的软件开发项目。该项目以html5为核心技术路线，致力于研究和实践程序设计和软件开发。本人通过广泛查阅相关技术书籍和文献，设计并开发了一个个性化的用户界面应用程序。在开发过程中，综合运用了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。不仅如此，项目还采用了响应式设计编程，适应了移动互联网时代用户屏幕多样化的需求。同时，采用了面向对象的程序设计思想，构建了一个通用的pointer模型，实现了对鼠标和触屏的控制。这一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，确保了项目的设计开发和测试质量。此外，在代码的开源和分享方面，项目团队利用git工具进行版本管理，并将代码上传到GitHub上，使得该应用程序可以便捷地跨平台高效访问

### **1. 前言**

我的毕设题目旨在探讨在Web UI设计领域中如何通过“读好书、练思维、勤编程”的方式提升自我的的专业能力和创造力。本论文将围绕这一主题展开深入研究，在不断学习和实践中如何不断完善自己。

#### 1.1毕设任务分析

随着信息技术的不断发展和普及，Web UI设计在传播信息、提升用户体验方面发挥着关键作用。因此，本次毕业设计将集中在Web UI设计领域展开，通过深入研究和实践，探讨如何运用“读好书、练思维、勤编程”的理念提升设计能力。

#### 1.2 研学计划

第1-3周：项目启动

目标：了解项目背景和需求，明确设计目标和关键要素。

任务：梳理项目需求，确定设计方向和风格，制定项目计划和时间表。

第4-5周：信息架构设计

目标：建立清晰的信息结构，确立页面布局和内容组织。

任务：进行线框图设计、网站地图规划等工作，为后续设计提供框架。

第6-10周：界面设计阶段

目标：设计视觉美感的界面。

任务：进行UI界面设计、配色方案制定、视觉元素搭配等工作，注重细节。

第11-15周：开发与测试

目标：将设计转化为可视化产品，进行功能开发和测试。

任务：完成UI界面的开发和测试工作，修复bug和优化体验。

第16周：项目总结与反思

目标：总结项目经验和教训，为未来设计工作提供借鉴。

任务：撰写项目总结报告、收集用户反馈、记录设计问题和解决方案。

通过严格按照以上16周的研学计划进行工作，能够在项目中有条不紊地推进工作，确保设计质量和进度控制，同时也提升自己的设计能力和实践经验。

#### 1.3研究方法

阅读与学习

书籍阅读：阅读与UI设计相关的经典著作和最新研究成果，提升对设计理论和实践的认识。

在线课程学习：仔细认证听教师讲解，系统学习UI设计和实现的操作与各种细节。认真完成课堂以及课后任务与作业，积极参与小组的项目编写。

### 2. 技术总结和文献综述

#### 2.1 Web平台和客户端技术概述

Web平台和客户端技术是互联网领域不可或缺的两大技术支柱。Web平台技术致力于构建基于Web的软件平台，实现跨平台、实时性、灵活性和安全性等特点；客户端技术则专注于构建并在用户设备上运行的应用程序，具备原生性、性能优势、离线访问和优质用户交互等特征。随着互联网和移动设备的广泛普及，Web平台和客户端技术在不断发展，包括前端技术更新、云服务兴起、移动化趋势和人工智能应用等方面的发展趋势。这些技术的不断创新和演进，旨在为用户提供更便捷、高效和丰富的应用体验，预示着未来互联网行业的潜力与挑战。

Web之父 Tim Berners-Lee在发明Web的基本技术架构以后，就成立了W3C组织，该组织在2010年后推出的HTML5国际标准，结合欧洲ECMA组织维护的ECMAScript国际标准，几乎完美缔造了全球开发者实现开发平台统一的理想，直到今天，科学家与Web行业也还一直在致力于完善这个伟大而光荣的理想[1]。学习Web标准和Web技术，学习编写Web程序和应用有关工具，最终架构一套高质量代码的跨平台运行的应用，是我的毕设项目应用的技术路线。

**2.2.1 History**

In 1989, [Sir Tim Berners-Lee](https://www.w3.org/People/Berners-Lee/) invented the World Wide Web (see the [original proposal](https://www.w3.org/History/1989/proposal.html)). He coined the term "World Wide Web," wrote the first World Wide Web server, "httpd," and the first client program (a browser and editor), "WorldWideWeb," in October 1990.

He wrote the first version of the "HyperText Markup Language" (HTML), the document formatting language with the capability for hypertext links that became the primary publishing format for the Web. His initial specifications for URIs, HTTP, and HTML were refined and discussed in larger circles as Web technology spread.

**2.2.2 A Consortium for the World Wide Web**

In 1994, the decision to form the World Wide Web Consortium came at the urging of many companies investing increasing resources into the web. Sir Tim Berners-Lee started leading the essential work of the Web Consortium team to foster a consistent architecture accommodating the rapid pace of progress in web standards for building websites, browsers, devices to experience all that the web has to offer.

In founding the World Wide Web Consortium, Sir Tim Berners-Lee created a community of peers. Web technologies were already moving so quickly that it was critical to assemble a single organization to coordinate web standards. Tim accepted the offer from MIT, who had experience with consortia, to host W3C. He required from the start that W3C have a global footprint.

**2.2.3 Web platform and Web Programming**

Let’s start with a brief description of the Web, which is short for World Wide Web. Most people say “Web” instead of “World Wide Web,” and we’ll follow that convention. The Web is a collection of documents, called web pages, that are shared (for the most part) by computer users throughout the world. Different types of web pages do different things, but at a minimum, they all display content on computer screens. By “content,” we mean text, pictures, and user input mechanisms like text boxes and buttons.[2]

Web programming is a large field, with different types of web programming implemented by different tools. All the tools work with the core language, HTML, so almost all the web programming books describe HTML to some extent. This textbook covers HTML5, CSS, and JavaScript, all in depth. Those three technologies are known to be the pillars of client-side web programming. With client side web programming, all web page calculations are performed on end users’ computers (the client computers).[3]

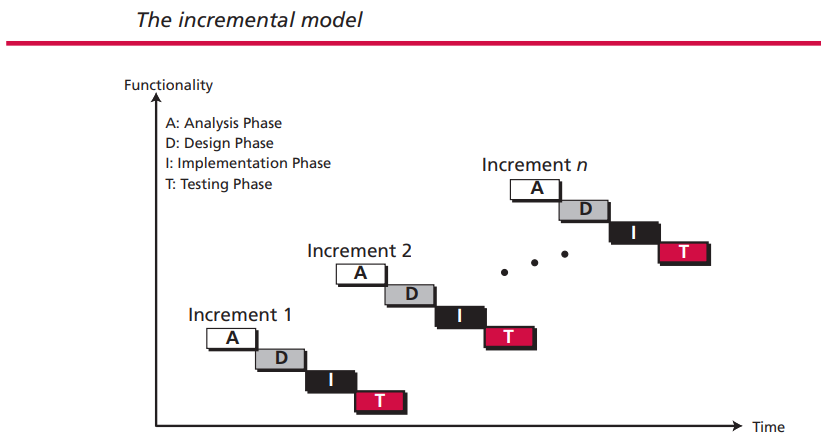
Web应用的程序设计体系由三大语言有机组成：HTML, CSS,  JavaScript。这三大语言的组合也体现了人类社会化大生产分工的智慧，可以看作用三套相对独立体系实现了对一个信息系统的描述和控制，可以总结为：HTML用来描述结构（Structure）、CSS用来描述外表（presentation）、Javascript用来描述行为（Behavior）[3]；这也可以用经典的MVC设计模式来理解Web平台架构的三大基石，Model可以理解为HTML标记语言建模，View可以理解为用CSS语言来实现外观，Controller则可理解为用 JavaScript结合前面二个层次，实现了在微观和功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

#### 2.2 项目的增量式迭代开发模式

本项目作为一个本科专业学生毕业设计的软件作品，与单一用途的程序相比较为复杂，本项目所涉及的手写代码量远超过简单一二个数量级以上，从分析问题的到初步尝试写代码也不是能在几天内能落实的，可以说本项目是一个系统工程，因此需要从软件工程的管理视角来看待和规范项目的编写过程。

而本项目考虑选择的软件工程开发过程管理模式有两种经典模型：瀑布模型（The waterfall model）和增量式迭代模型(The incremental model)。而任何开发模式则都必须同样经历四个阶段：分析（Analysis）、设计（Design）、实施（Implementation）、测试（test）。

瀑布模型需要专业团队完美的配合，从分析、设计到实施，最后到测试，任何阶段的开始必须基于上一阶段的完美结束。而这对于我们大多数普通开发者是不太现实的，作为小微开发者由于身兼数职，其实无法1次就能完美完成任何阶段的工作，比如在实施过程中，开发者会发现前面的设计存在问题，则必须在下一次迭代项目时改良设计。在当今开源的软件开发环境中，开发者在软件的开发中总是在不断地优化设计、重构代码，持续改进程序的功能和代码质量。因此在本项目的开发中，也采用了增量模型的开发模式[5]。本项目中我一共做了六次项目的开发迭代，如下图2-1所示：



The incremental model

In the incremental model, software is developed in a series of steps. The developers first complete a simplified version of the whole system. This version represents the entire system but does not include the details. Figure shows the incremental model concept.

In the second version, more details are added, while some are left unfinished, and the system is tested again. If there is a problem, the developers know that the problem is with the new functionality. They do not add more functionality until the existing system works properly. This process continues until all required functionality has been added. [5]

### 3. 内容设计概要

#### 3.1 分析和设计

这一步是项目的初次开发，本项目最初使用人们习惯的“三段论”式简洁方式开展内容设计，首先用一个标题性信息展示logo或文字标题，吸引用户的注意力，迅速表达主题；然后展现主要区域，也就是内容区，“内容为王”是项目必须坚守的理念，也是整个UI应用的重点；最后则是足部的附加信息，用来显示一些用户可能关心的细节变化。如图4-1用例图所示：



标题区

内容区

附加信息区

Web UI 应用

**图4-1用例图**

#### 3.2 项目的实现和编程

**一、HTML代码编写如下：**

<header>

《 我的毕设题目 》

</header>

<main>

我的主题内容： ‘读好书、练思维、勤编程’ @masterLijh 计算思维系列课程

</main>

<footer>

CopyRight XXX 江西科技师范大学 2024-2025

</footer>

**二、CSS代码编写如下：**

\*{

margin: 10px;

text-align: center;

font-size:30px ;

}

header{

border: 2px solid blue;

height: 200px;

}

main{

border: 2px solid blue;

height: 400px;

}

footer{

border: 2px solid blue;

height: 100px;

}

a{

display: inline-block ;

padding:10px ;

color: white;

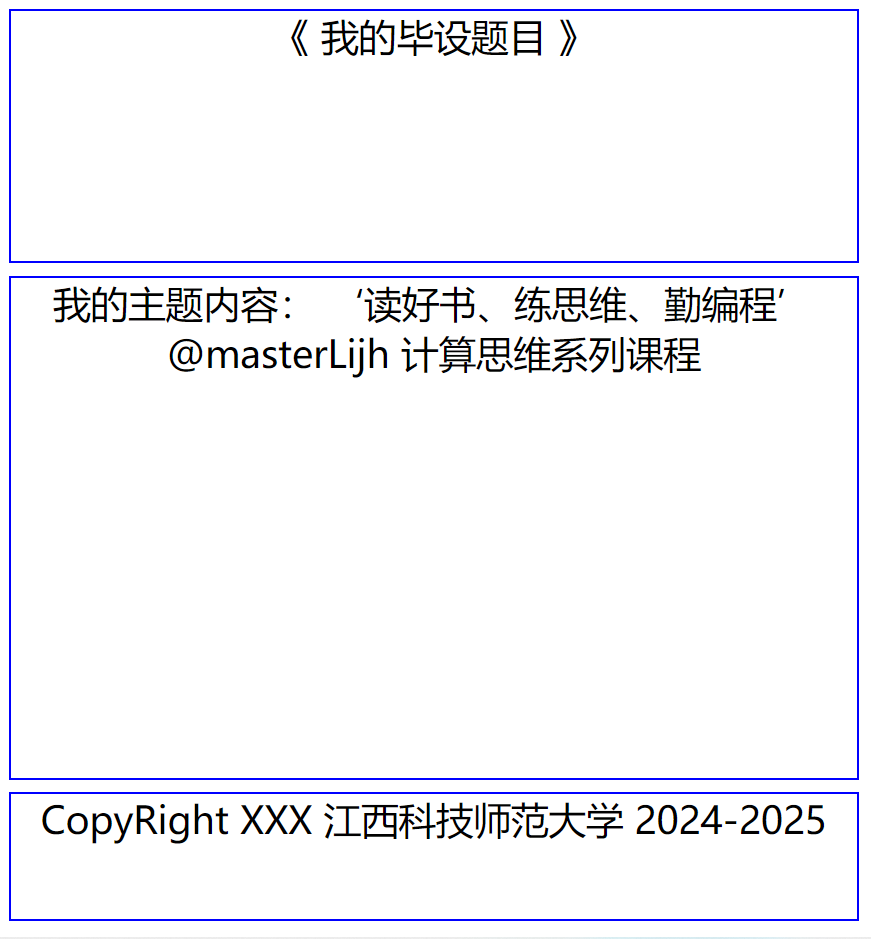
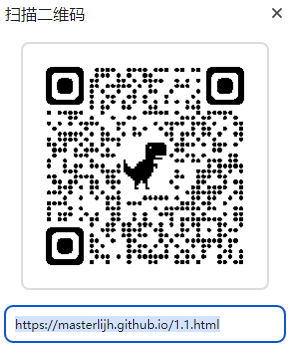
background-color: blue;

text-decoration: none ;

}

#### 3.3 项目的运行和测试

项目的运行和测试至少要通过二类终端，本文此处仅给出PC端用Chrome浏览器打开项目的结果，如下图4-2所示。由于本项目的阶段性文件已经上传github网站，移动端用户可以通过扫描图4-3的二维码，运行测试本项目的第一次开发的阶段性效果。

**图4-2 PC端运行效果图 图4-3 移动端二维码**

#### 3.4 项目的代码提交和版本管理

本项目的文件通过gitBash工具管理，作为项目的第一次迭代，在代码提交和版本管理环节，我们的目标是建立项目的基本文件结构，还有设置好代码仓库的基本信息：如开发者的名字和电子邮件。

进入gitBash命令行后，按次序输入以下命令：

$ cd /

$ mkdir webUI

$ cd webUI

$ git init

$ git config user.name 江科师大李健宏

$ git config user.email marsterlijh@jxstnu.edu.cn

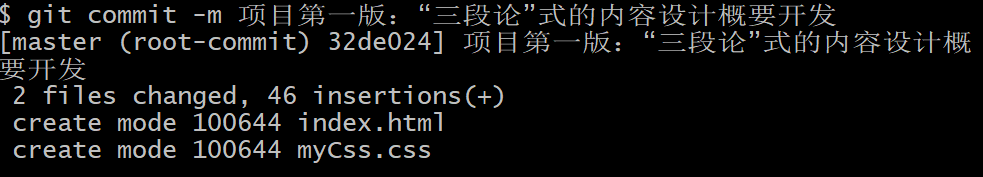
$ touch index.html myCss.css

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

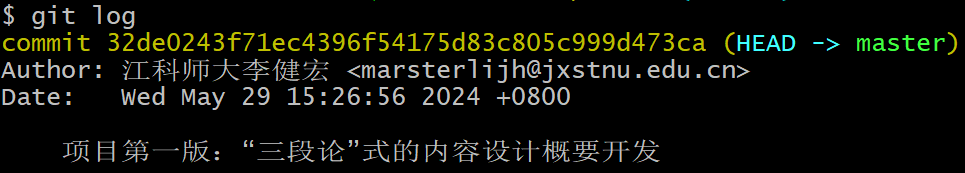
成功提交代码后，gitbash的反馈如下所示：



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：



### 移动互联时代的UI开发初步——窄屏终端的响应式设计

随着移动互联网的迅速发展，用户越来越多地通过各种移动设备访问网络内容。窄屏终端，如智能手机和平板电脑，已经成为用户与互联网互动的主要工具之一。因此，响应式设计成为Web UI开发中不可或缺的一部分，它能够确保网站无论在何种屏幕尺寸和分辨率下都能提供良好的用户体验。

**Responsive Design--Accommodating Display Hardware**

The display hardware used with computers varies widely, with the size and resolution of a display depending on cost. Rather than have a version of each web page for each type of display, designers chose to make web pages give general layout guidelines and allow a browser to choose how to display the page on a given computer. Thus, a web page does not give many details. For example, the author of a web page can specify that a group of sentences form a paragraph, but the author cannot specify details such as the exact length of a line or whether to indent the beginning of the paragraph.[1]

Allowing a browser to choose display details has an interesting consequence: a web page may appear differently when viewed through two browsers or on two computers that have dissimilar hardware. If one screen is wider than another, the length of a line of text or the size of images that can be displayed differs.The point is: A web page gives general guidelines about the desired presentation; a browser chooses details when displaying a page. As a result, the same web page can appear slightly different when displayed on two different computers or by different browsers.[1]

分析移动互联时代的多样化屏幕的需求。

用JavaScript开动态读取显示设备的信息，然后按设计，使用js+css来部署适配当前设备的显示的代码。

实现代码

用汉语言来描述我们是如何实现的，与上一阶段比较，本阶段初次引入了em和 % ，这是CSS语言中比较高阶的语法，可以有效地实现我们的响应式设计 。如代码块4-1所示：

<style>

\*{

margin: 10px;

text-align: center;

}

header{

border: 2px solid blue;

height: 15%;

font-size: 1.66em;

}

main{

border: 2px solid blue;

height: 70%;

font-size: 1.2em;

}

nav{

border: 2px solid blue;

height: 10%;

}

nav button{

font-size: 1.1em;

}

footer{

border: 2px solid blue;

height: 5%;

}

</style>

代码块4-1

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。如代码块4-2所示：

<script>

var UI = {};

UI.appWidth = window.innerWidth > 600 ? 600 : window.innerWidth ;

UI.appHeight = window.innerHeight;

const LETTERS = 22 ;

const baseFont = UI.appWidth / LETTERS;

//通过更改body对象的字体大小，这个属性能够遗传其子子孙孙

document.body.style.fontSize = baseFont + "px";

//通过把body对象的宽度和高度设置为设备/屏幕的宽度和高度，实现全屏。

//通过CSS对子对象百分比（纵向）的配合，从而实现响应式设计的目标。

document.body.style.width = UI.appWidth - 2\*baseFont + "px" ;

document.body.style.height = UI.appHeight - 4\*baseFont + "px";

</script>

代码块4-2

### 应用响应式设计技术开发可适配窄屏和宽屏的UI

随着移动设备的普及和多样化，用户对于能够在不同尺寸和分辨率的屏幕上提供一致体验的网站需求日益增长。为了满足这一需求，现代Web UI设计趋向于采用响应式设计技术，以确保网站可以适配窄屏和宽屏设备上的展示效果。

在本章中，我们将探讨如何应用响应式设计技术来开发一个能够在窄屏和宽屏设备上提供良好用户体验的UI。

响应式设计原则

流动布局：使用流动布局可以让UI元素根据屏幕大小自动调整位置和大小，从而适应不同设备的显示效果。

媒体查询：利用媒体查询功能，可以根据不同的屏幕尺寸应用不同的CSS样式，以实现针对性的布局调整。

弹性图片和媒体：为了适配不同分辨率的屏幕，可以使用弹性图片和媒体资源，以保证在任何设备上都能够显示清晰且不失真。

技术实现

设置Viewport：通过设置Viewport meta标签，可以告诉浏览器如何缩放页面以适应不同设备的屏幕宽度。

媒体查询：编写不同屏幕尺寸下的媒体查询规则，根据设备宽度的不同调整UI元素的样式，比如改变文本大小、布局结构等。

Flexbox布局：使用Flexbox布局可以轻松地实现弹性且多列的页面布局，适配不同尺寸的屏幕。

图片优化：针对不同分辨率的屏幕加载不同尺寸的图片，避免在窄屏设备上加载过大的图片导致页面加载缓慢。

用户体验考量

触摸友好：为窄屏设备优化用户交互方式，确保按钮大小合适，避免拥挤的布局影响用户点击体验。

内容重要性：在窄屏设备上，需要考虑内容的优先级，确保关键信息能够在较小的屏幕空间内清晰展示。

动态效果：利用CSS3动画和过渡效果，增强用户在窄屏设备上的交互体验，但注意不要过度使用影响页面加载速度。

通过以上的应用响应式设计技术，开发适配窄屏和宽屏的UI界面将能够为用户提供更加统一和一致的浏览体验，并能够在不同设备上展现出色的用户界面设计。

### 个性化UI设计中对鼠标交互的设计开发

个性化UI设计在当前Web开发中扮演着越来越重要的角色，它可以增强用户体验、提升品牌形象，而对鼠标交互设计的精心开发是其中不可或缺的一环。在本章中，我们将探讨如何利用CSS和JavaScript技术为个性化UI设计中的鼠标交互注入活力和创意。

创意鼠标指针设计

自定义鼠标指针样式：通过CSS的cursor属性，可以自定义不同样式的鼠标指针，如改变形状、颜色或添加动画效果，从而为用户提供独特的鼠标交互体验。

定制化鼠标图标：利用CSS和SVG技术，可以创建个性化的鼠标图标，比如根据不同状态改变鼠标形状，增加悬停效果或定制特殊动画。

交互效果增强

鼠标悬停效果：通过CSS的hover伪类和过渡效果，为UI元素添加悬停交互效果，比如改变颜色、透明度或放大缩小，以吸引用户的注意力。

鼠标点击动画：利用JavaScript技术，为鼠标点击事件添加动画效果，如点击按钮时产生涟漪效果或按钮变形，增加用户操作的反馈感。

用户体验优化

减少鼠标移动次数：设计UI时考虑用户使用习惯，尽量减少鼠标移动次数，通过合理的布局和交互设计来提高用户效率。

鼠标响应速度：优化界面响应速度，确保鼠标指针移动和点击操作的流畅性，避免延迟和卡顿现象。

定制化交互体验

根据网站主题定制鼠标设计：将网站主题与鼠标设计相融合，创造统一的视觉风格，增加品牌辨识度。

个性化交互动效：结合鼠标设计和动效，打造独特的用户交互体验，通过与用户的互动增强网站的吸引力和趣味性。

通过以上的个性化UI设计中对鼠标的精心设计和开发，能够为用户带来更加独特和愉悦的交互体验，提升用户对于网站的满意度和粘性。同时，合理地运用鼠标交互设计也能够帮助网站实现品牌定位、提升用户留存率，是Web UI设计中不可忽视的重要环节。

### 对触屏和鼠标的通用交互操作的设计开发

随着移动设备的普及和多样化，Web UI设计必须考虑到不同操作方式下用户体验的差异，尤其是触屏和鼠标的操作。在本章中，我们将探讨如何设计和开发能够同时满足触屏和鼠标操作的通用性UI界面。

设计原则

简洁直观：在设计UI时，应保持界面简洁直观，避免过多复杂操作，使用户能够快速找到所需功能。

响应式布局：采用响应式设计，确保UI界面能够根据不同设备尺寸和操作方式自动调整布局，保证在触屏和鼠标操作下都能提供良好的用户体验。

触屏操作设计

提供大目标体验：在触屏设备上，用户通过手指进行操作，因此按钮和交互元素的大小要足够大，以增加点击命中率。

手势操作支持：考虑加入常见手势操作，如滑动、捏合和旋转等，以提高用户对于界面的操作灵活性。

鼠标操作设计

准确指向：鼠标操作需要准确的指向，因此按钮和链接的大小适中，避免操作失误。

悬停效果：利用鼠标悬停效果（hover）提供更多信息或视觉提示，增强用户操作的反馈感。

通用操作设计

统一交互逻辑：确保在触屏和鼠标操作下的交互逻辑一致，用户无论使用何种设备都能够直观地使用界面功能。

交互反馈：针对不同操作方式提供明确的交互反馈，如按钮点击效果、页面加载状态提示等，帮助用户更好地理解其操作结果。

用户体验优化

用户测试：进行用户测试，根据用户反馈不断优化UI设计，以确保用户能够流畅地使用网站功能。

性能优化：优化页面加载速度和响应速度，提高用户体验，特别是在触屏设备上更要关注性能优化。

通过对触屏和鼠标的通用操作进行设计开发，可以为用户提供更加流畅、直观且一致的操作体验。合理地结合触屏和鼠标操作的设计原则，能够提升网站的用户满意度和互动性，从而达到提升用户留存和忠诚度的目的

### UI的个性化键盘交互控制的设计开发

键盘作为人机交互的重要方式，在Web UI设计中扮演着重要角色。本章将讨论如何利用键盘事件来实现个性化交互控制，并探索未来键盘功能的发展方向。

键盘事件的基本原理

在Web开发中，键盘事件主要分为两类：keydown和keyup。keydown事件在键盘按键被按下时触发，keyup事件在按键被释放时触发。通过监听这些事件，可以获取用户键盘操作的相关信息，并相应地进行交互控制。

设计与开发思路

为实现UI的个性化键盘交互控制，设计与开发思路如下：

事件监听：通过JavaScript监听keydown和keyup事件，捕获用户的键盘操作。

交互反馈：根据用户按键的不同，设计相应的交互反馈，提高用户体验。

功能扩展：探索键盘操作的更多潜力，结合UI设计的特点拓展更丰富的交互功能。

响应式设计：针对不同设备的键盘输入习惯进行设计，保证在不同平台上的交互效果一致。

底层事件的应用

通过探索和利用keydown和keyup键盘底层事件，可以为UI的键盘功能提供更强大的潜力。例如，可以实现以下功能：

快捷键操作：设计自定义快捷键，提高用户操作效率。

游戏交互：实现游戏键盘控制功能，增强游戏互动体验。

多媒体控制：结合键盘事件控制音视频播放，提供更便捷的操作方式。

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)

});

### 谈谈本项目中的高质量代码

This is a book about instructing computers. Computers are about as common as screwdrivers today, but they are quite a bit more complex, and making them do what you want them to do isn’t always easy.

If the task you have for your computer is a common, well-understood one, such as showing you your email or acting like a calculator, you can open the appropriate application and get to work. But for unique or open-ended tasks, there probably is no application.

That is where programming may come in. Programming is the act of constructing a program--a set of precise instructions telling a computer what to do. Because computers are dumb, pedantic beasts, programming is fundamentally tedious and frustrating. Fortunately, if you can get over that fact, and maybe even enjoy the rigor of thinking in terms that dumb machines can deal with, programming can be rewarding. It allows you to do things in seconds that would take forever by hand. It is a way to make your computer tool do things that it couldn’t do before. And it provides a wonderful exercise in abstract thinking.[6]

创建一个Pointer对象，践行MVC设计模式，设计一套代码同时对鼠标和触屏实现控制。

面向对象思想，封装，抽象，局部变量，函数式编程，逻辑。

（围绕着抽象定义函数、代码块、模型设计以及降低全局变量的使用来写）

### 用gitBash工具管理项目的代码仓库和http服务器

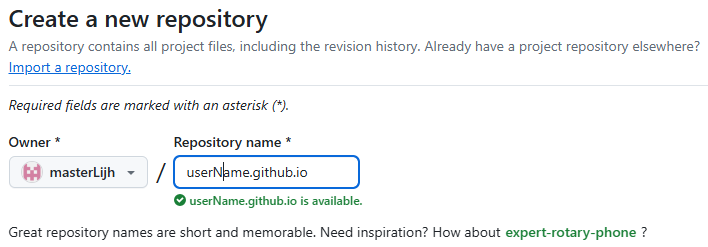
#### 10.1 经典Bash工具介绍

When we speak of the command line, we are really referring to the shell. The shell is a program that takes keyboard commands and passes them to the operating system to carry out. Almost all Linux distributions supply a shell program from the GNU Project called bash. The name is an acronym for bourne-again shell, a reference to the fact that bash is an enhanced replacement for sh, the original Unix shell program written by Steve Bourne.[7]

Like Windows, a Unix-like operating system such as Linux organizes its files in what is called a hierarchical directory structure. This means they are organized in a tree-like pattern of directories (sometimes called folders in other systems), which may contain files and other directories. The first directory in the file system is called the root directory. The root directory contains files and sub directories, which contain more files and sub directories, and so on.[7]

#### 10.2 通过gitHub平台实现本项目的全球域名

#### 10.3 创建一个空的远程代码仓库





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

#### 10.4 设置本地仓库和远程代码仓库的链接

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

$ echo "WebUI应用的远程http服务器设置" >> README.md

$ git init

$ git add README.md

$ git commit -m "这是我第一次把代码仓库上传至gitHub平台"

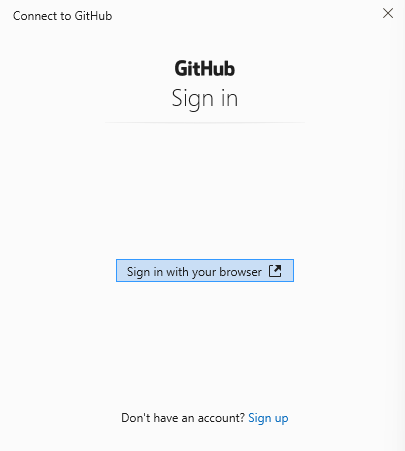
$ git branch -M main

$ git remote add origin

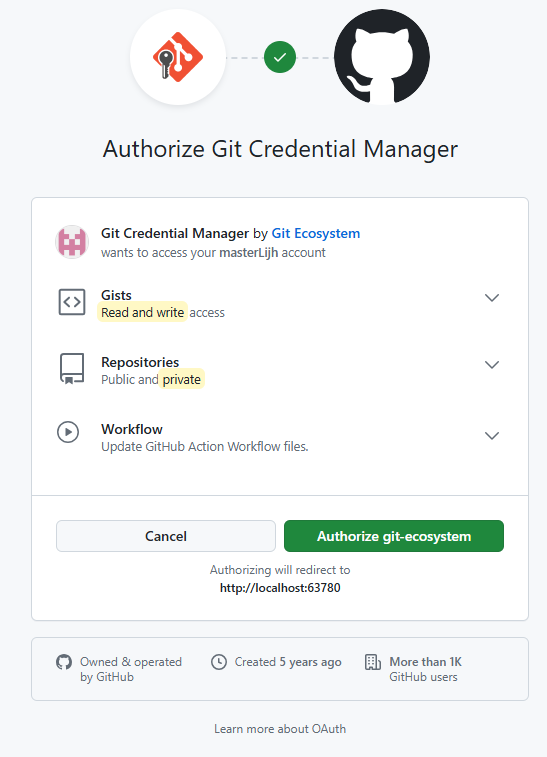
<https://github.com/masterLijh/userName.github.io.git>

$ git push -u origin main

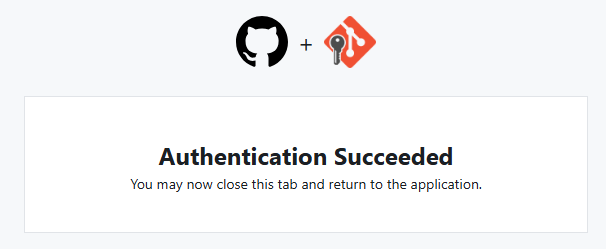
本项目使用window平台，gitbash通过默认浏览器实现密钥生成和记录，第一次链接会要求开发者授权，如下图所示：



再次确认授权gitBash拥有访问改动远程代码的权限，如下图所示：



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



从此，我们无论在本地做了任何多次代码修改，也无论提交了多少次，上传远程时都会把这些代码和修改的历史记录全部上传github平台，而远程上传命令则可简化为一条：git push ，极大地方便了本Web应用的互联网发布。

远程代码上传后，项目可以说免费便捷地实现了在互联网的部署，用户可以通过域名或二维码打开，本次使用PC的微软Edge浏览器打开，本文截取操作中间的效果图，如下所示：



全文完成，谢谢！

### 参考文献：

1. W3C. W3C's history. W3C Community. [EB/OL]. https://www.w3.org/about/. <https://www.w3.org/about/history/.> 2023.12.20
2. Douglas E. Comer. The Internet Book [M] (Fifth Edition). CRC Press Taylor & Francis Group, 2019: 217-218
3. John Dean,PhD. Web programming with HTML5,CSS,and JavaScript[M]. Jones & Bartlett Learning,LLC. 2019: 2
4. John Dean,PhD. Web programming with HTML5,CSS,and JavaScript[M]. Jones & Bartlett Learning,LLC. 2019: xi
5. Behrouz Forouzan. Foundations of Computer Science[M](4th Edition). Cengage Learning EMEA,2018: 274--275
6. Marijn Haverbeke. Eloquent JavaScript 3rd edition. No Starch Press,Inc, 2019.
7. William Shotts. The Linux Command Line, 2nd Edition [ M ]. No Starch Press, Inc, 245 8th Street, San Francisco, CA 94103, 2019：3-7

### 写作指导：

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

【摘要案例1】近十年来，html5为核心的web标准的软件开发技术以其跨平台、开源的优势广泛地运用在各个领域的应用软件开发中。通过分析本次毕设任务，本项目选择html5的web客户端技术为技术路线，展开对程序设计和软件开发的研究和实践。通过广泛查阅相关技术书籍、开发者论坛和文献，设计开发了一个个性化的用户界面（UI）的应用程序。在开发中综合应用了html语言进行内容建模、css语言展开UI的外观设计、javascript语言编程实现UI的交互功能，除直接使用了web客户端最底层的API外，本项目的每条代码都是手工逐条编写，没有导入他人的任何的代码（框架和库）。本项目也采用了响应式设计编程，可以智能地适应移动互联网时代用户屏幕多样化的需要；另外大量地运用了面向对象的程序设计思想，比如用代码构建了一个通用的pointer模型，该模型仅用一套代码就实现了对鼠标和触屏的控制，实现了高质量的代码，这也是本项目的亮点。从工程管理的角度看，本项目采用的增量式开发模式，以逐步求精的方式展开了六次代码的增量式重构（A:Analysis，D:Design，I：Implementation，T:Testing）,比较愉快地实现项目的设计开发和测试。从代码的开源和分享的角度看，本项目采用了git工具进行版本管理，在漫长的开发过程中重构代码六次并正式做了代码提交，另外在测试中修改提交了代码两次，最后利用gitbash工具 把本项目的代码仓库上传到著名的github上，再利用github提供的http服务器，本项目实现了UI应用在全球互联网的部署，我们可以通过地址和二维码便捷地跨平台高效访问这个程序。

【摘要案例2】：Web技术以其跨操作系统平台的优势成为了广泛流行的软件开发手段，为了适应移动互联网时代软件项目的前端需求，本项目以Web客户端技术为研究学习内容，广泛查阅了技术资料与相关文献，尤其是mozilla组织的MDN社区的技术实践文章，探索了HTML内容建模、CSS样式设计和JavaScript功能编程的基本技术和技巧。通过集成上述技术，再应用本科的相关课程的知识，实现了一个个性化的用户界面（UI：uer interface）的项目, 该用户界面以响应式技术为支撑做到了最佳适配用户屏幕，程序可以动态适用于当前PC端和移动端设备；在功能上以DOM技术和事件驱动模式的程序为支撑实现了对鼠标、触屏、键盘的底层事件响应和流畅支持，为鼠标和触屏设计了一个对象模型，用代码实现了对这类指向性设备的模拟（这是本项目模型研究法的一次创新实践，也是本项目的亮点。）。为了处理好设计和开发的关系，项目用了工程思想管理，使用了软件工程的增量式开发模式，共做了6次项目迭代开发，每次迭代都经历了开发4个经典开发阶段（A:Analysis，D:Design，I：Implementation，T:Testing），以逐步求精的方式编写了本UI的应用程序。为了分享和共享本代码，与网上的开发者共同合作，本项目还使用了git工具进行代码和开发过程日志记录，一共做了12次提交代码的操作，详细记录和展现了开发思路和代码优化的过程，最后通过gitbash把项目上传到github上，建立了自己的代码仓库，并将该代码仓库设置成为了http服务器，实现了本UI应用的全球便捷访问。