

目录

基于 Web 客户端技术的个性化 UI 的设计和实现	2
(Customized UI design and Programming based on Web client technology)	2
1. 前言	2
1.1 毕设任务分析	2
1.2 研学计划	3
1.3 研究方法	3
2. 技术总结和文献综述	4
2.1 Web 平台和客户端技术概述	4
2.2 项目的增量式迭代开发模式	5
3. 内容设计概要	7
3.1 分析和设计	7
3.2 项目的实现和编程	7
3.3 项目的运行和测试	8
3.4 项目的代码提交和版本管理	9
4. 移动互联时代的 UI 开发初步——窄屏终端的响应式设计	10
5. 应用响应式设计技术开发可适配窄屏和宽屏的 UI	12
6. 个性化 UI 设计中对鼠标交互的设计开发	12
7. 对触屏和鼠标的通用交互操作的设计开发	12
8. UI 的个性化键盘交互控制的设计开发	12
9. 谈谈本项目中的高质量代码	13
10. 用 gitBash 工具管理项目的代码仓库和 http 服务器	14
10.1 经典 Bash 工具介绍	14
10.2 通过 gitHub 平台实现本项目的全球域名	14
10.3 创建一个空的远程代码仓库	15
10.4 设置本地仓库和远程代码仓库的链接	15
参考文献:	19
写作指导:	19

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 李登金扬

摘要：个性化用户界面（UI）在 Web 应用程序中的设计和实现是当前研究的焦点之一。随着互联网的发展和用户需求的多样化，传统的静态和固定 UI 已经不能满足用户的需求。个性化 UI 通过根据用户的偏好、行为和上下文信息来动态调整界面，提升了用户体验和系统的可用性。本文从设计和实现两个方面探讨了基于 Web 客户端技术的个性化 UI 的关键问题。设计方面，个性化 UI 的实现涉及用户数据的收集、分析和利用。通过收集用户的历史操作数据、偏好设置以及实时的行为数据，系统可以建立用户模型，并据此推断用户的兴趣和需求。基于这些信息，系统可以自动调整界面元素的展示顺序、样式和内容，从而优化用户的交互体验。实现方面，个性化 UI 的技术支持主要包括前端页面的动态生成和响应式设计。随着前端技术的不断发展，如 Ajax、JavaScript 框架（如 React、Angular 等）和 CSS 媒体查询，使得在不同终端设备上实现响应式布局变得更加容易和高效。同时，利用服务器端的数据分析和处理，可以实现个性化推荐和动态内容生成，进一步提升用户的满意度和粘性。基于 Web 客户端技术的个性化 UI 设计和实现在提升用户体验、增强系统互动性方面具有重要意义。未来的研究方向包括如何更精准地分析用户行为、如何提高个性化推荐的准确性和实时性，以及如何在保护用户隐私的前提下有效利用用户数据等问题。个性化 UI 的发展将继续推动 Web 应用程序设计和开发的进步，为用户提供更加智能、高效的交互体验。

关键词：个性化用户界面；JavaScript 框架；css；web；

1. 前言

在当今数字化社会中，Web 应用程序已经成为人们日常生活中不可或缺的一部分。随着互联网技术的迅猛发展和普及，用户对 Web 应用的个性化需求日益增加。个性化用户界面（UI）的设计和实现因其能够根据用户偏好和行为动态调整而备受关注。本文旨在探讨基于 Web 客户端技术的个性化 UI 的设计和实现，以满足用户对个性化体验的需求。在传统的 Web 应用中，用户界面通常是静态和普遍适用的，无法有效地适应不同用户群体的个性化需求。然而，随着信息技术的发展和用户体验设计的深入研究，个性化 UI 成为提升用户满意度和参与度的重要手段之一。个性化 UI 不仅能够根据用户的兴趣、偏好和使用习惯实时调整界面元素，还能提供更加精准和符合个体需求的服务和内容。本文将集中讨论如何利用现代 Web 客户端技术（如 HTML5、CSS3、JavaScript 等）设计和实现个性化 UI。通过分析用户行为数据、偏好信息以及实时反馈机制，我们可以构建一种智能化的用户界面，使其能够根据用户的动态需求和上下文环境进行实时调整和优化。这不仅有助于提升用户体验的个性

化程度，还能增强应用程序的用户粘性和市场竞争力。

1.1 毕设任务分析

毕业设计（毕设）任务分析是确保项目成功完成的关键步骤。首先，需要明确项目的背景和目的，包括所属学科领域及研究背景。其次，详细分析任务要求和约束条件，如功能、性能、时间等方面的具体要求和限制。然后，界定项目的范围和限制，确保项目不偏离主题。接着，确定关键任务和阶段性目标，有助于分解复杂任务，更有效地推进项目。最后，制定合理的项目计划和时间安排，评估风险并做好资源规划和管理，以确保项目顺利进行。这些步骤和考虑因素将为毕设的顺利完成提供必要的指导和支持。

1.2 研学计划

在计算机科学的毕业设计项目和毕业论文中，制定一个有效的学习计划至关重要。明确研究领域和课题选择，首先要确定自己感兴趣的研究领域和具体的课题。这可以通过阅读相关文献、参加课程和讨论来达成。学习和研究背景知识，确保掌握和理解与选定课题相关的基础知识和理论。这可能涉及到深入学习特定的编程语言、算法、数据结构、数据库技术等。制定时间表和阶段性目标，制定一个详细的时间表，包括每个阶段的任务和目标。这有助于确保项目在规定时间内按时完成，并有助于分配资源和管理时间。持续学习和反馈机制，毕业设计项目应该是一个持续学习的过程。定期审查进展，并寻求指导教师或导师的反馈和建议，以确保项目在正确的轨道上进行。

1.3 研究方法

有效的研究方法是确保毕业设计项目和毕业论文质量的关键。以下是几种常见的研究方法：

（1）实证研究：如果课题适合实证研究，可以设计实验或调查来收集和分析数据，验证假设或解决实际问题。

（2）文献综述：进行系统性的文献综述，总结和分析前人的研究成果和观点，为自己的研究提供理论依据和背景。

（3）案例研究：选择一个典型的案例或实际应用进行深入分析和研究，探讨其在特定环境下的应用效果和影响。

（4）原型开发和评估：如果项目涉及软件开发，可以采用原型开发方法，快速迭代并收集用户反馈，以改进系统的设计和功能。

（5）数学建模和仿真：对于某些技术或理论，可以采用数学建模和仿真方法来验证理论模型的有效性和准确性。

2. 技术总结和文献综述

2.1 Web 平台和客户端技术概述

Web 之父 Tim Berners-Lee 创立了 Web 的核心技术框架，并建立了 W3C 组织。自 2010 年以来，W3C 推出的 HTML5 国际标准与 ECMA 国际组织维护的 ECMAScript 标准的结合，几乎完美地实现了全球开发者追求统一开发平台的梦想。至今，科学家和 Web 行业依然不懈努力，致力于进一步完善这一伟大而光荣的愿景[1]。

在我的毕设项目中，我专注于深入学习和应用 Web 标准及相关技术，以掌握编写跨平台运行的高质量 Web 应用所需的工具和技能。我的目标是打造一套优秀的代码架构，能够在不同平台上无缝运行的应用程序。

2.2.1 History

In 1989, Sir Tim Berners-Lee invented the World Wide Web (see the original proposal). 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 1 所示：

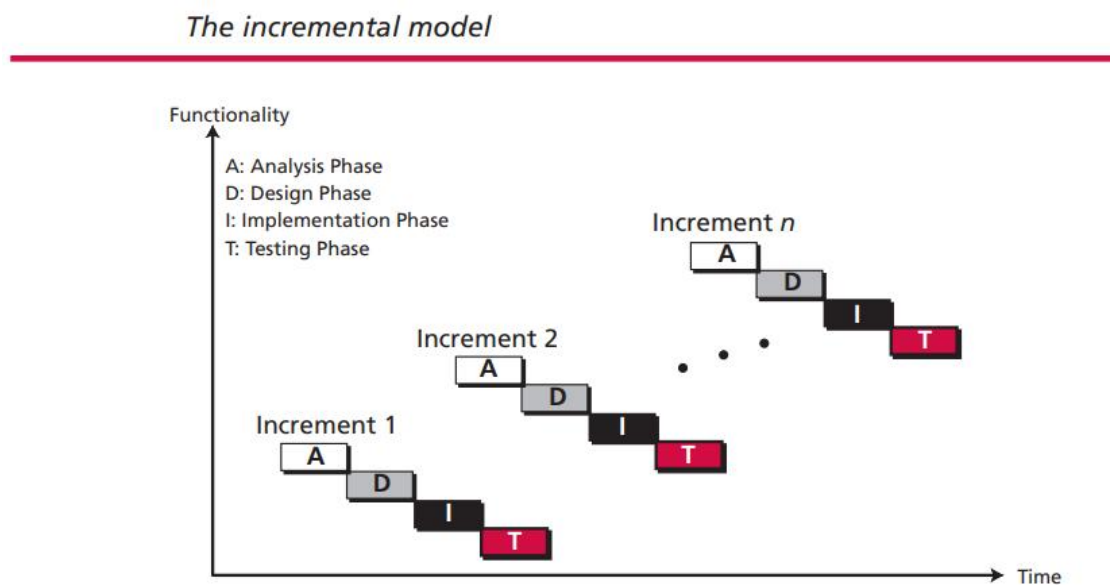


图 1

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：

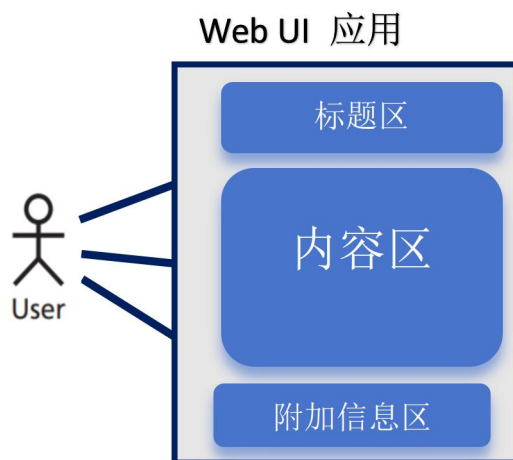


图 4-1 用例图

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

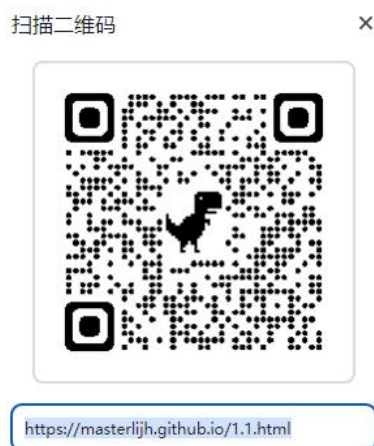


图 4-3 移动端二维码

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 /d/abc
$ mkdir webUI
$ cd webUI
$ git init
$ git config user.name 科师大李登金扬
$ git config user.email 2539686415@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
```

c

```
git commit -m 项目第一版：“三段论”式的内容设计概要开  
in 7d1828c] 项目第一版：“三段论”式的内容设计概要开  
file changed, 1 insertion(+), 1 deletion(-)
```

项目代码仓库自此也开启了严肃的历史记录，我们可以输入日志命令查看，

```
$ git log
```

gitbash 反馈代码的仓库日志如下所示：

```
commit 7d1828c2f4aaa1658c727c784450c5c0bfb249e2 (HEAD -  
> main)  
Author: 科师大李登金扬 <2539686415@qq.com>  
Date:   Wed Jun 19 13:50:09 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

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

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

6. 个性化 UI 设计中对鼠标交互的设计开发

阐述用一套代码逻辑同时为触屏和鼠标建立对象模型

7. 对触屏和鼠标的通用交互操作的设计开发

阐述用一套代码逻辑同时为触屏和鼠标建立对象模型

8. UI 的个性化键盘交互控制的设计开发

阐述探索和利用 keydown 和 keyup 键盘底层事件，为未来 UI 的键盘功能提供底层强大的潜力。

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)
});

```

9. 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）

10. 用 gitBash 工具管理项目的代码仓库和 http 服务器

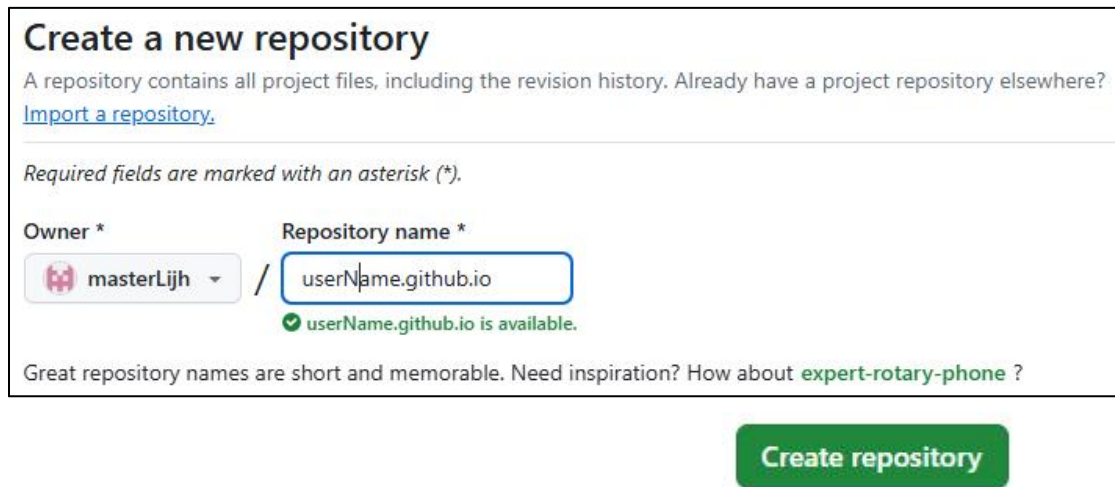
10.1 经典 Bash 工具介绍

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




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

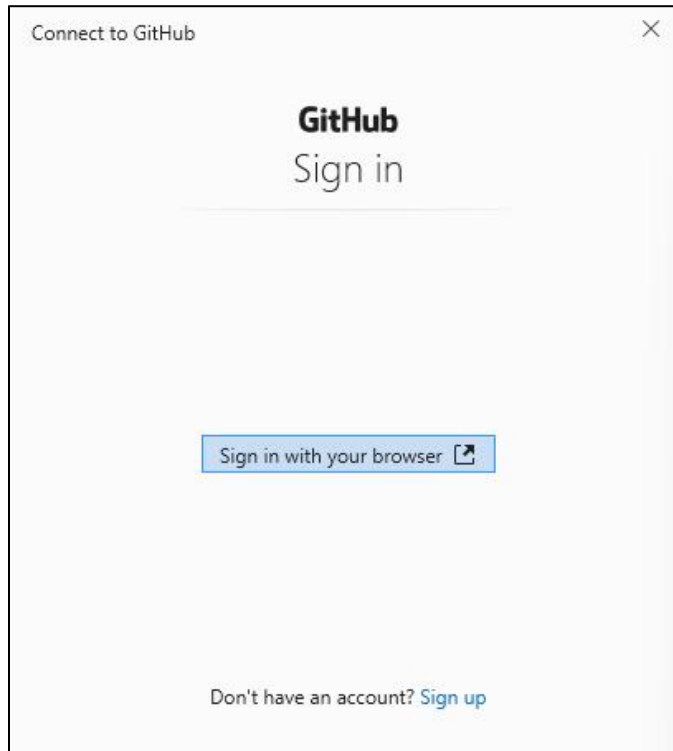
点击窗口右下角的绿色“Create repository”，则可创建一个空的远程代码仓库。

10.4 设置本地仓库和远程代码仓库的连接

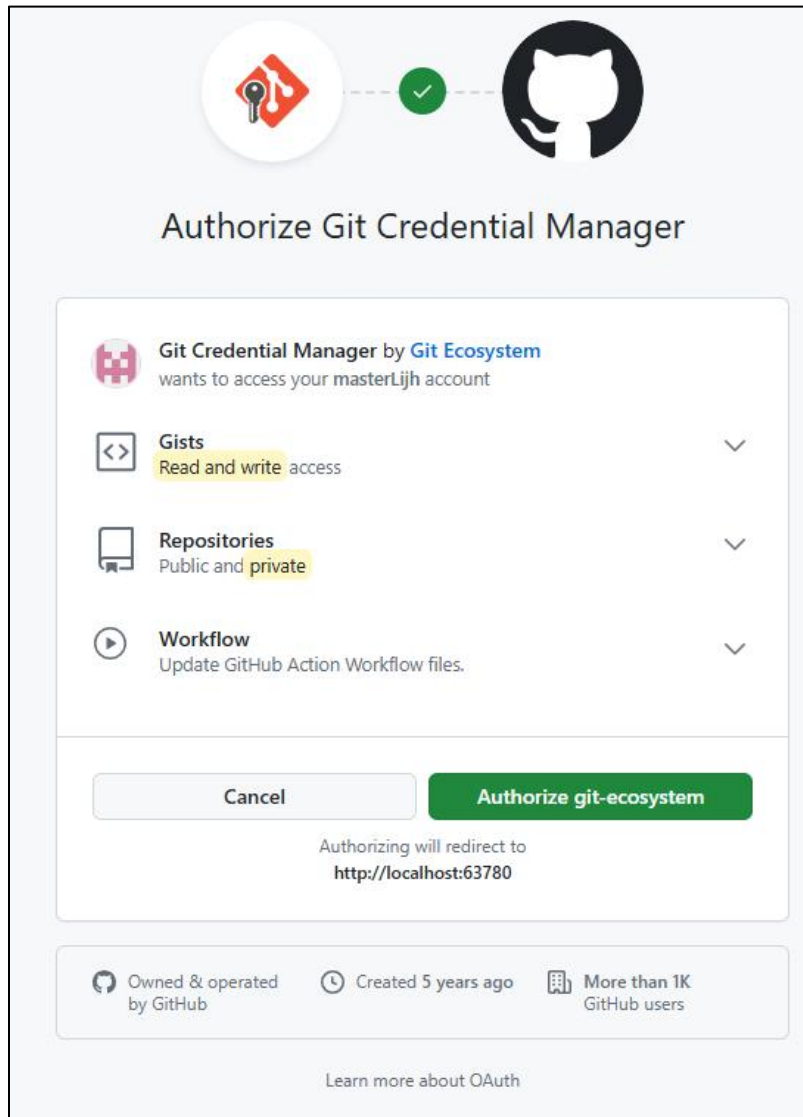
进入本地 webUI 项目的文件夹后，通过下面的命令把本地代码仓库与远程建立密钥链接

```
$ echo "WebUI 应用的远程 http 服务器设置" >> README.md
$ git init
$ git add README.md
$ git commit -m "这是我第一次把代码仓库上传至 gitHub 平台"
$ git branch -M main
$ git remote add origin
    https://github.com/masterLijh/userName.github.io.git
$ git push -u origin main
```

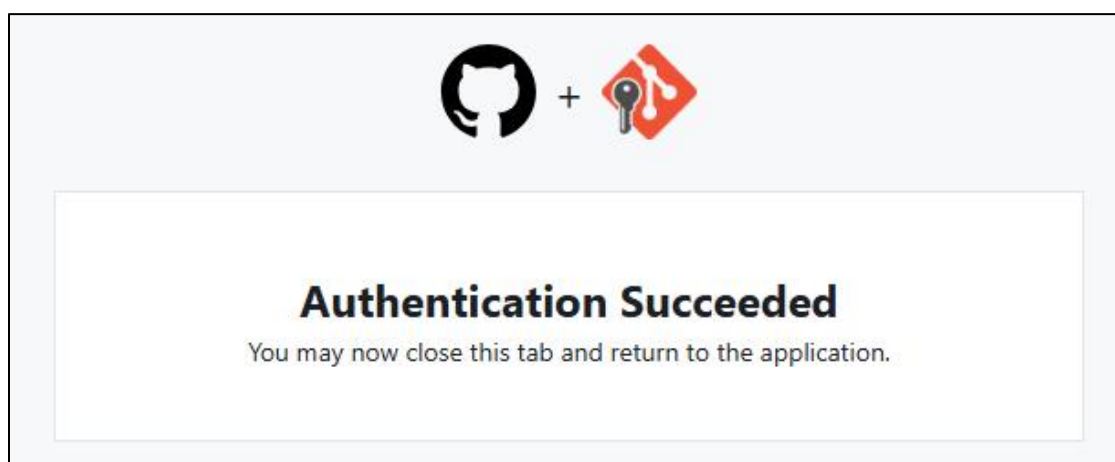
本项目使用 window 平台，gitbash 通过默认浏览器实现密钥生成和记录，第一次链接会要求开发者授权，如下图所示：



再次确认授权 `gitBash` 拥有访问改动远程代码的权限，如下图所示：

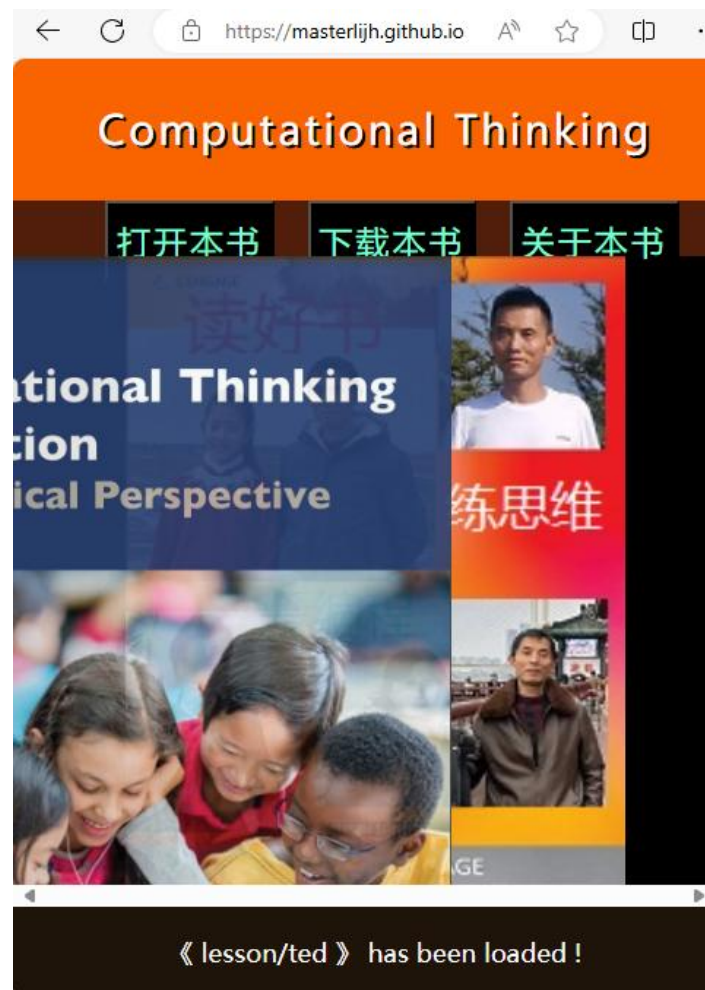


最后，GitHub 平台反馈：gitBash 和 gitHub 平台成功实现远程链接。



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

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



全文完成，谢谢！

参考文献：

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

写作指导：

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

【摘要案例 1】近十年来，html5 为核心的 web 标准的软件开发技术以其跨平台、开源的优势广泛地运用在各个领域的应用软件开发中。通过分析本次毕设任务，本项目选择 html5 的 web 客户端技术为技术路线，展开对程序设计和软件开发的研究和实践。通过广泛查阅相关技术书籍、开发者论坛和文献，设计开发了一个个性化的用户界面（UI）的应用程序。在开发中综合应用了 html 语言进行内容建模、css 语言展开 UI 的外观设计、javascript 语言编程实现 UI 的交互功能，除直接使用了 web 客户端最底层的 API 外，本项目的每条代码都是手工逐条编写，没有导入他人的任何的代码（框架和库）。本项目也采用了响应式设计编程，可以智能地适应移动互联网时代用户屏幕多样化的需要；另外大量地运用了面向对象的程序设计思想，比如用代码构建了一个通用的 pointer 模型，该模型仅用一套代码就实现了对鼠标和触屏的控制，实现了高质量的代码，这也是本项目的亮点。从工程管理的角度看，

本项目采用的增量式开发模式，以逐步求精的方式展开了六次代码的增量式重构（A:Analysis, D:Design, I: Implementation, T:Testing），比较愉快地实现项目的设计开发和测试。从代码的开源和分享的角度看，本项目采用了 git 工具进行版本管理，在漫长的开发过程中重构代码六次并正式做了代码提交，另外在测试中修改提交了代码两次，最后利用 gitbash 工具 把本项目的代码仓库上传到著名的 github 上，再利用 github 提供的 http 服务器，本项目实现了 UI 应用在全球互联网的部署，我们可以通过地址和二维码便捷地跨平台高效访问这个程序。

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