

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 的设计和编程

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

院（系）：元宇宙产业学院

专 业：计算机科学与技术

学生姓名：尹乐欣

学 号：20213922

指导教师：李健宏

2024 年 6 月 2 日

目录

| | |
|--|----|
| 1.引言 | 1 |
| 1.1 项目概述 | 2 |
| 1.2 研学计划 | 2 |
| 1.3 研究方法 | 2 |
| 2.Web 平台和客户端技术概述 | 4 |
| 2.1Web 平台 | 4 |
| 2.2Web 编程 | 4 |
| 2.3 软件开发的过程管理——增量式开发模式 | 5 |
| 3.内容设计概要 | 6 |
| 3.1 分析和设计 | 7 |
| 3.2 项目的实现和编程 | 8 |
| 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 分析和设计 | 11 |
| 4.2 项目的实现和编程 | 11 |
| 4.3 项目的运行和测试 | 13 |
| 5.移动互联时代的响应式设计和窄屏代码实现 | 14 |
| 5.1 分析和设计 | 14 |
| 5.2 项目的实现和编程 | 14 |
| 5.3 项目的运行和测试 | 16 |
| 6.适用移动互联时代的响应式设计 | 17 |
| 6.1 分析和设计 | 17 |
| 6.2 项目的实现和编程 | 17 |
| 7.个性化 UI 设计中鼠标模型以及通用 UI 设计 | 21 |
| 7.1 分析和设计 | 21 |
| 7.2 项目的实现和编程 | 22 |
| 8.UI 的个性化键盘控制 | 27 |
| 8.1 分析和设计 | 27 |
| 8.2 项目的实现和编程 | 27 |
| 9.本项目中的高质量代码 | 30 |
| 10.用 gitBash 工具管理项目的代码仓库和 http 服务器 | 31 |
| 10.1 经典 Bash 工具介绍 | 31 |
| 10.2 通过 github 平台访问项目 | 31 |
| 参考文献 | 32 |

基于 Web 客户端技术的个性化 UI 的设计和实现

摘要：近十年来，以 HTML5 为核心的网络标准因其跨平台、开源等优势，受到了越来越多的关注。本课题基于 Web 客户端技术，对其编程与软件开发进行了深入探索与实践。经过深入研究相关技术，并参与开发团队，最终完成了一款基于用户界面的个性化应用。在系统设计中，采用基于 Web 客户端的 API 接口，使用 HTML、CSS、JavaScript 等技术完成内容生成、接口设计及交互功能设计。为了适应移动互联时代对终端设备多样性的需求，本课题采用了响应式设计策略。此外，引入面向对象的设计思想，利用统一代码库构建指针模型，使其兼容多种输入设备，提升代码质量与可维护性。在项目管理层面，采用渐进式开发方式，通过反复优化与改进，完成产品分析、设计、实现、测试等多个阶段。对于开源与代码分享，我们使用 Git 进行版本控制，对代码多次重构与修改，最终上传到 GitHub 平台。通过 GitHub 的 HTTP 协议进行部署，让用户只需通过网页连接或扫描二维码即可轻松接入，实现跨平台的便捷接入。

关键词：Web 客户端技术；计算机科学技术；网页；HTML

1.引言

在 Web 客户机技术基础上进行个性化用户接口设计，是一个涉及多个领域的综合学习过程。首先，学习 HTML、CSS、JavaScript，并逐步过渡到更高层次的前端架构如 React、Vue、Angular，再学习 CSS 预处理和前端编译工具。深入了解前端技术后，开始个性化用户界面元件的设计与开发，并掌握 Git 等版本管理工具。后续研究涉及后端技术、数据库技术和 API 设计。接着，研究如何利用机器学习提升用户体验，并通过数据分析优化设计。从需求分析入手，确定项目目的与用户需求，选择适合的前后端技术，完成产品原型和草图，然后进行前后端开发，包括数据处理和商业逻辑实现。综合测试确保前后端无缝整合，性能优化集中在载入时间和响应能力。项目结束后，需投入使用并根据用户反馈不断改进。参考材料如《HTML & CSS: Design and Build Websites》、《JavaScript: The Good Parts》及 MDN、W3Schools、FreeCodeCamp 等网络资源是重要的学习资料。设计工具如 Adobe XD、Sketch、Figma 对用户界面设计有帮助。通过文献回顾了解已有设计与实施方式，个案分析提供实际应用参考。本文提出了一种基于软件开发的原型系统，研究表明数据分析方法能够更好地理解用户习惯和喜好，机器学习技术能够提供个性化体验。项目实施将形成一套完整的用户界面设计与程序设计理论体系，并开发出兼具美感与实用性的用户界面应用程序。

1.1 项目概述

利用截屏和文字两种方法,对计算机专业软件的毕业论文进行了验证。另外,本研究也可将毕设作品的网址或二维码提供给读者,让读者可以直观地感受并利用其作品。这样既能提高文章的可信性,又能突出单纯靠文字描述很难做到的特点。本专业主要课程包括: OOP 语言,数据结构和算法,操作系统,软件工程等。由于其强调的是理论性的内容,所以在以前的教学中,这些理论都是比较抽象的。经过与老师的交流,大家都同意,如果能将本专业的主要内容运用于实际操作,则能从认知与技能两个层次上提高专业水准。对电脑开发人员来说,对电脑系统的深入了解是非常重要的,因为它不但是未来的工程师们所要具备的,也是他们区别于其它专业的地方。从其它专业的观点来看,计算机专业的学生对计算机体系的认识不能只停留在表层,而应该更加深入地了解它的实质,在技术上也要尽量贴近它的基础。

1.2 研学计划

毕设的完工可分为两个阶段。第一阶段是确定技术实务路径、掌握核心技术、查阅指导老师的个案研究、了解各技术之间的内在联系以及它们在项目中的角色与分工。本课程强调通过项目执行来提高代码质量。在此基础上,理论与实践相结合,进入第二阶段,即完成毕设软件的研制。此阶段按照软件工程规范,包括以下步骤:定义与分析问题,设计技术方案,编写程序与代码,执行技术部署,调试程序,进行软件测试与性能优化。在之前的步骤中,我们可能会发现一些问题,这些问题会导致我们在设计、编码、测试等方面反复修改,以纠正设计上的错误。本文的主要工作是建立编码系统并实施。

1.3 研究方法

在我的毕业设计中,我尝试解决的问题是按照层次和类别进行界定的,这种方法帮助我更系统地组织和分析问题。模型研究方法是一种抽象的研究方法,它不依赖于特定的计算机语言,而是通过理论模型来分析和解决问题。在系统设计过程中,我选择使用统一建模语言(UML)对系统进行建模。我相信,从目的

上看，UML 模型研究方法与面向对象编程（OOP）方法在本质上是相同的，尽管它们在问题分析和形成观点的方式上有所不同。在我的毕业设计中，我使用 UML 来建立问题的模型。UML 是一种相对抽象的语言，这在设计过程中确实带来了不少困难。通过对面向对象编程语言的学习与应用，我能够对模型进行运行与调试，进而发现模型设计中存在的问题和漏洞。这种方法不仅提高了我的编程技能，还增强了我对模型的理解和修正能力。对于计算机科学专业的本科学生来说，可以通过编写程序进行建模和研究。在完成程序设计后，再使用 UML 绘制出相应的模型，并将其撰写成一份详细的文档。毕竟，大学生们的思维尚在成熟过程中，他们不像资深的编程人员或系统架构师那样，具备丰富的抽象能力和经验。通过这种方法，学生们能够逐步培养自己的抽象思维和系统设计能力。在学术研究中，文献法是一种基本且重要的研究方法。它以对前人作品的研读和分析为核心，帮助研究者深入了解某一学科的知识 and 技巧。不论你处于程序设计的哪个阶段，文献方法都是网络应用程序开发专业学生必不可少的一部分。知识存在于多种形式中，如书籍、在线文档、社区论坛、学术期刊、会议报告等。通过这些途径，学生们能够获取大量的信息和资源，从而提升自己的知识水平和研究能力。另外，UML 作为一种标准化的建模语言，不仅用于软件工程，还广泛应用于其他工程领域，例如系统工程、业务流程建模等。它包括一系列图表，用来描述系统的静态结构和动态行为。通过学习和使用 UML，我不仅掌握了如何进行系统建模，还增强了对复杂系统的理解能力。这为我未来从事系统设计和开发打下了坚实的基础。在我的毕业设计过程中，我遇到了许多挑战，但这些挑战也带来了宝贵的经验和学习机会。例如，在使用 UML 建模时，我需要反复进行模型的设计和调整，以确保模型的准确性和有效性。这不仅锻炼了我的耐心和细致的工作态度，也提高了我的问题解决能力和创新思维。总的来说，通过我的毕业设计，我深刻体会到模型研究方法和 UML 在系统设计中的重要性。它们不仅帮助我更好地理解 and 解决问题，还为我提供了一套系统的分析和设计工具。通过不断的学习和实践，我相信自己能够在未来的职业生涯中，应用这些方法和工具，解决更加复杂和多样化的问题。因此，不论是在学术研究还是实际工作中，掌握模型研究方法和 UML 都是至关重要的。这不仅有助于提高我们的专业技能，还能增强我们的竞争力和职业发展前景。通过不断的学习和积累经验，我们可以不断

提升自己的能力，为成为一名优秀的系统设计师或软件工程师奠定坚实的基础。

2.Web 平台和客户端技术概述

2.1Web 平台

Web 开发是一个内容广阔的技术领域。那么，何为 Web 呢？根据百度百科的解释，Web（World Wide Web）即全球广域网，也称为万维网。它是一种基于超文本和 HTTP 的、全球性的、动态交互的、跨平台的分布式图形信息系统。Web 建立在 Internet 之上，为浏览者提供了在 Internet 上查找和浏览信息的图形化、易于访问的直观界面。通过文档及超级链接，Web 将 Internet 上的信息节点组织成一个互为关联的网状结构。当我们在手机或电脑浏览器上输入网址（学名为 URL，即 Uniform Resource Locator，统一资源定位符）时，就能进入图形化的交互页面。例如，我们可以在某度上搜索资料，或在某宝上下单购买一双鞋。通常，我们平常接触到的是前端，即与用户交互的界面。然而，Web 开发不仅仅包括前端，还包括后端和数据库，每个方面都涉及广泛的知识与技术。以某宝为例，平台上有用户、商家和维护人员。不同的角色对应不同的界面与权限。比如，商家上架商品后，这些商品需要及时呈现在用户端；而管理员如果发现某商铺违反了平台规定，可以立即关停该商铺。这些功能和权限的管理都可以理解为后端设计的一部分。通过前端、后端和数据库的协同工作，Web 开发实现了复杂的功能和流畅的用户体验，支持了现代互联网的繁荣发展。

2.2Web 编程

Web 编程设计是一门庞大的学科，使用各种工具来实现网络程序。然而，几乎每一种工具都以 HTML 为核心语言。网络应用的编程体系主要由 HTML、CSS 和 JavaScript 三种语言构成。它们的结合反映了人类社会大生产的智慧，可以看作是用三种相对独立的体系来描述和控制一个信息系统：HTML 用于描述结构，CSS 用于描述外观，JavaScript 用于描述行为。这可以用传统的 MVC 设计模式来理解：模型（Model）是 HTML 标签语言的建模，视图（View）由 CSS 实现，

控制器（Controller）通过 JavaScript 将前两个层次结合，从而在微观和功能上对代码进行控制。

2.3 软件开发的过程管理——增量式开发模式

软件开发过程一般分为四个阶段，每个阶段都有具体工作和重点。这四个阶段包括需求分析、设计、开发和测试。目前已有多种软件生命周期模型，各有特点、优缺点及应用范围。使用不同模式的软件产品，其生命周期各不相同。在此基础上，提出了一种面向对象的软件开发方法，并进行了分析。这里，我们仅探讨两种最常用的软件开发模式。

2.3.1 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 2.1 所示。

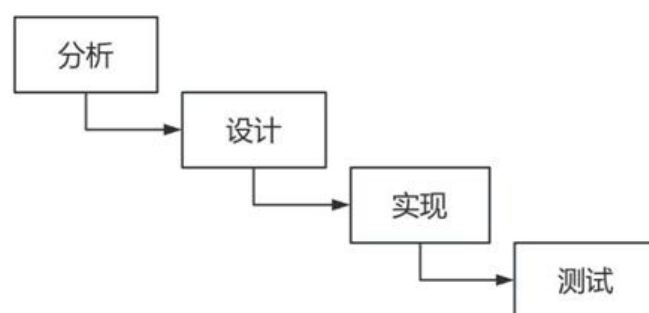


图 2.1 瀑布模型

2.3.2 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。增量式模

式的显著特征是模块化、构件化。增量式模型正是利用了这一特性。增量模型如图 2.2 所示。

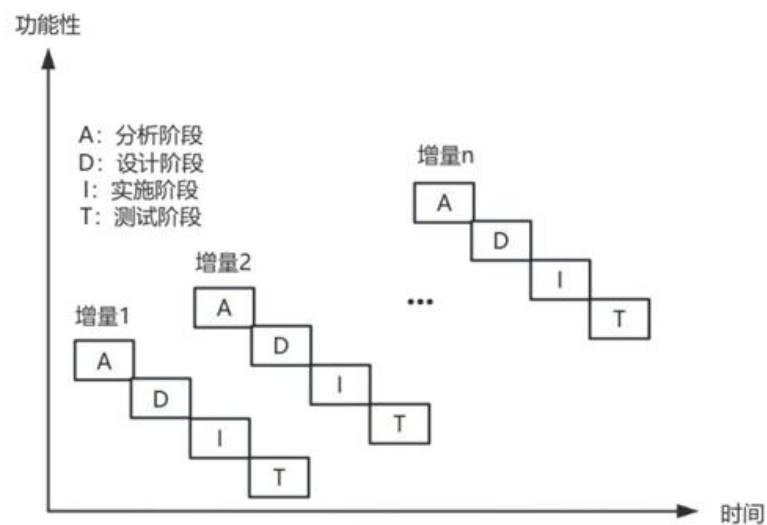


图 2.2 增量模型

3.内容设计概要

在资讯科技快速发展的今天，网络已经渗透到我们的日常生活中。针对当前网络环境下日益提高的个性化、交互性要求，本课题拟研究面向 Web 应用的高交互性、高交互性。本文的研究背景是在对当前网络应用进行分析的基础上得出的结论：用户对界面的美感、易用性以及响应速度的要求越来越高。为此，本课题旨在设计和实现一款兼具美感与实用性的 Web 应用，从而提高用户体验，更好地适应市场需求。整个项目的蓝图截图如图 3.1 所示。



图 3.1 项目蓝图示意图

3.1 分析和设计

本项目采用了经典的“三段论”模式，通过明确的结构和层次来传递信息，旨在确保用户体验的流畅和舒适。首先，项目以醒目的标题作为起点，吸引用户的注意力并引导他们进一步探索内容。这一部分的设计强调视觉冲击力和信息的简洁性，以便快速抓住用户的兴趣点。接下来，主要内容区域作为整个设计的核心部分，包含了高质量、高价值的内容。这里的信息经过精心策划和优化，旨在与用户进行深度互动。通过富有吸引力的文本和图像，以及精心布置的多媒体元素，这一部分不仅提供了丰富的信息，还鼓励用户参与和反馈，从而增强用户的参与感和满意度。最后，附加信息区域则用于展示细节和变化，进一步提升用户体验和信任度。这一部分可能包括更新日志、用户评价、FAQ 等内容，旨在提供更多背景信息和支持，帮助用户更全面地理解和使用产品或服务。通过这种全面、细致的设计，项目不仅增强了信息的传递效果，还提升了整体的用户体验，使用户在每一个接触点都感受到专业和关怀。如图 3.2 所示。

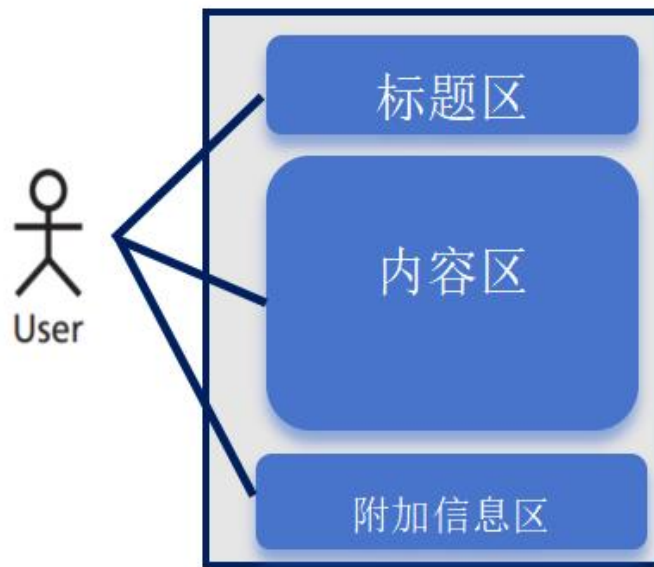


图 3.2 项目用例图

3.2 项目的实现和编程

3.2.1 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>
```

3.2.2CSS 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

在测试与验证阶段, 我将这段代码上传到 [github](#) 上并通过浏览器生成了该网页的二维码, 在电脑端上的效果图如图 3.3 所示。

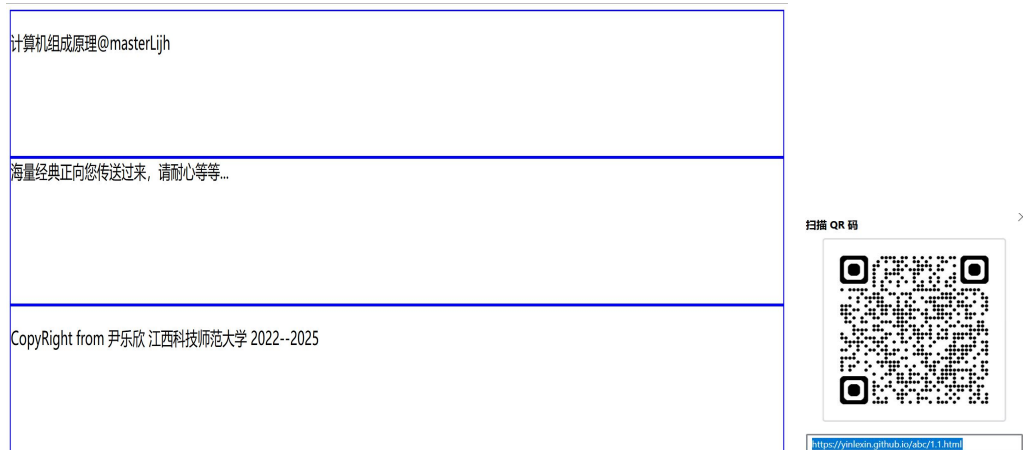


图 3.3 PC 端运行效果图

3.4 项目的代码提交

本项目的文件管理采用 gitBash 工具，旨在首次迭代中确立项目的基础文件架构，并完成代码仓库的初步设置，包括录入开发者的姓名与电子邮件地址等基本信息。在启动 gitBash 的命令行界面后，需依次执行以下指令序列：

```
$ cd /
$ mkdir webUI
$ cd webUI
$ git init
$ git config user.name yinlexin
$ git config user.email 1094012048@qq.com
$ touch index.html myCss.css
```

完成 index.html 和 myCss.css 文件的编码工作，并在测试确认无误后，进行以下步骤以提交代码：

```
$ git add index.html myCss.css
$ git commit -m 提交了 1.1.html 和基础样式...
```

提交后的日志如图 3.4 所示。

```
commit f6fa7f603ca3f9206b013ea36cdf5a17f63aacca (HEAD -> master)
Author: yinlexin <1094012048@qq.com>
Date: Wed Jun 19 18:59:55 2024 +0800
```

提交了1.1.html和基础样式...

图 3.4 提交效果图

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 分析和设计

分析移动互联时代用户终端的多样性对网页设计提出的要求，以及响应式设计的重要性和实现方式。利用了多种技术和工具，包括 CSS、JavaScript、CSS3、SVG、jQuery、响应式框架和库等，以实现网页的自适应性和适应性。这些工具可以根据屏幕尺寸的变化自动调整布局，优化图片媒体的使用，应用不同的样式规则，动态调整特定功能，并提供测试和验证机制来确保网页能够在不同设备和环境下提供良好的用户体验。通过这些技术的综合应用，可以构建出能够自适应不同用户终端的网页，满足移动互联时代对网页可访问性和视觉一致性的需求。

4.2 项目的实现和编程

```
<style>
    body {

        text-align: center;
    }

    header {
        border: 2px solid blue;
        height: 10%;
        background-color: blue;
        color: white;
    }
    nav {
        border: 2px solid blue;
        height: 5%;
    }
```

```

main {
    height: 75%;
    background-image: url(/lesson/CS.jpg);
    background-repeat: no-repeat;
    background-size: cover;
    background-position: center;
}

footer {
    border: 2px solid blue;
    height: 10%;
    background-color: black;
    color: white;
}

p#book {
    font-size: 1.8em;
    letter-spacing: 0.2em;
}

button {
    font-size: 1.2em;
    margin-right: 1em;
}

</style>

<script>
var UI = {};

UI.deviceWidth = window.innerWidth;

UI.deviceHeight = window.innerHeight;

document.body.style.height = UI.deviceHeight + 'px';

```

```
document.body.style.fontSize = UI.deviceWidth/25 + 'px';  
</script>
```

4.3 项目的运行和测试

该网页基于 HTML 构建，包括标题、导航、主要内容区域和页脚。视窗使用 `<meta>` 标签进行设置，以确保在不同设备上实现响应式布局。CSS 样式直接嵌入在 "`<style>`" 标签中，定义了网页元素的大小、颜色、背景图片和字体风格。背景图片通过 CSS 中的 "`background-size: cover`" 属性来覆盖整个主元素，确保不会扭曲图片。JavaScript 脚本动态调整字体尺寸，以提高网页的响应速度和可读性。页眉和页脚采用一致的设计风格，通过颜色和边框突出网页的视觉主题。网页还包括导航按钮，尽管没有提供具体的导航功能。页脚显示了标准的版权信息。总体来说，内联样式可以简化样式管理，但会影响样式的重用和维护。该代码结合了 HTML、CSS、JavaScript 等技术，展示了基本的响应能力。上传项目代码到 GitHub 后，即可获得相应的链接。

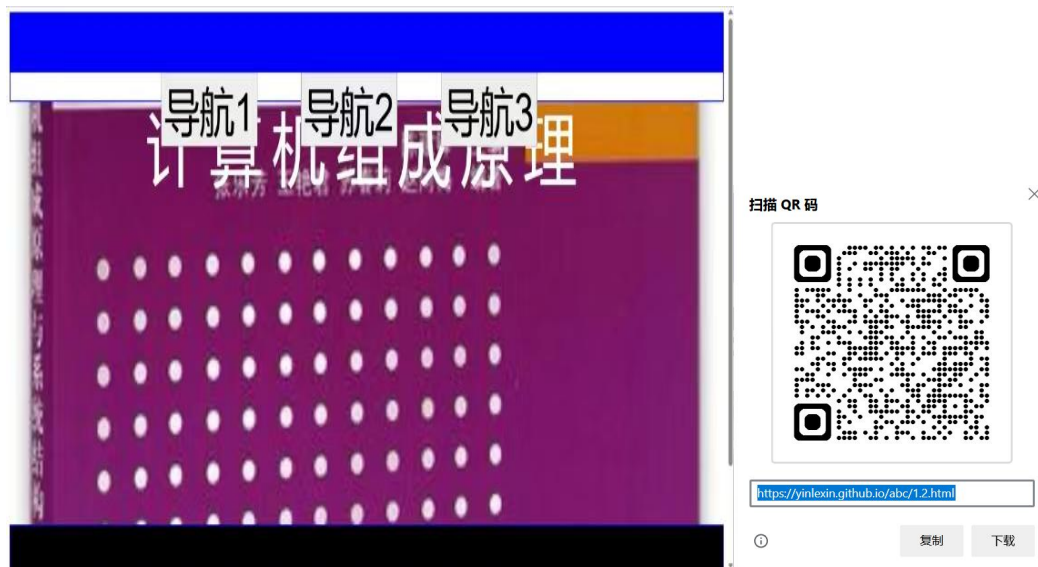


图 4.1 运行效果图

5.移动互联时代的响应式设计和窄屏代码实现

5.1 分析和设计

与上次相比，我对页面的 CSS 风格进行了更改。我使用选择器来选取网页中的全部元素，并设定预设的外边距和文字对齐方式。这样的设定通常用于确保网页元素之间有适当的间隔，并使文字内容居中显示。对于特定的网页元素，如标题、段落和主体内容，我定义了相应的边框、高度和字体尺寸。所有元素都使用了 3 像素的实心绿线作为边界，其高度可以根据网页布局进行指定。标题和段落各占网页高度的 10%，主体内容占 70%，而字体则占 10%。这是一种典型的页面布局形式。在主体部分，我使用了一个带有绝对定位的 #box 元素 (position: absolute)。该元素总是以最近的祖先元素为基准（如果没有，则以最初的包含块为准），并且其右边总是与所包含块的右边对齐。这样的定位方式可以让开发人员准确地将内容放置在主要区域中。

5.2 项目的实现和编程

具体代码如下所示：

```
<style>

*{
    margin: 10px;
    text-align: center;
}

header{
    border: 3px solid green;
    height: 10%;
    font-size: 1em;

}

nav{
```



```
border: 3px solid green;
height: 10%;
}
main{
border: 3px solid green;
height: 70%;
font-size: 0.8em;
position: relative;
}

#box{
position: absolute;
right: 0;
width: 100px;
}

footer{
border: 3px solid green;
height: 10%;
font-size: 0.7em;
}
body{
position: relative;
}
#aid{
position: absolute;
border: 3px solid blue;
top: 0px;
left: 600px;
}
```

```

#bookface{
    width: 80%;
    height: 80%;
    border:1px solid red;
    background-color: blanchedalmond;
    margin:auto;
}
</style>

```

5.3 项目的运行和测试

在这个应用程序中，我运用了一个定制功能和一个事件侦听器，以便捕捉并积极响应用户的鼠标动作。这一举措不仅使得用户体验更加流畅和直观，同时也奠定了未来互动设计的基石。通过这种技术手段，我们能够更好地满足用户需求，提升用户参与度，并为他们带来更加个性化和令人愉悦的交互体验。这种创新的设计理念将为用户带来全新的互动感受，为应用程序的发展注入了更多可能性和活力。运行结果如图 5.1 所示：



图 5.1 运行效果图

6.适用移动互联时代的响应式设计

6.1 分析和设计

数字时代的到来使得快速、个性化的网站设计变得至关重要，要根据用户需求进行灵活调整。随着移动设备成为主要上网工具，网站需要能够灵活适应不同尺寸和分辨率，以提供优秀用户体验。响应性设计可以确保网站在多种设备上呈现一致的视觉效果，让用户无缝浏览，提高满意度和忠诚度，并降低维护成本。相比传统的站点设计，响应性设计通过一套代码库实现多设备自适应，提升了维护效率。此外，响应性设计也有利于搜索引擎优化，因为搜索引擎更青睐能为手机用户提供出色体验的站点。响应性设计还能加速网页下载，通过资源优化降低页面大小，提高浏览速度，减少用户流失。统一的用户经验有助于建立客户对品牌的信任和忠诚度，提升企业形象和市场竞争力。

6.2 项目的实现和编程

具体代码如下所示：

```
<script>
  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' ;
    }
}

```

```

});

function $(ele){
    if (typeof ele !== 'string'){
        throw("自定义的$函数参数的数据类型错误，实参必须是字符串！");
    }

    let dom = document.getElementById(ele);

    if(dom){
        return dom;
    }else{
        dom = document.querySelector(ele);
        if (dom) {
            return dom;
        }else{
            throw("执行$函数未能在页面上获取任何元素，请自查问题！");
        }
    }

    return;
}

} //end of $

</script>

```

6.3 项目的运行与测试

项目的运行结果如图 6.1 所示：

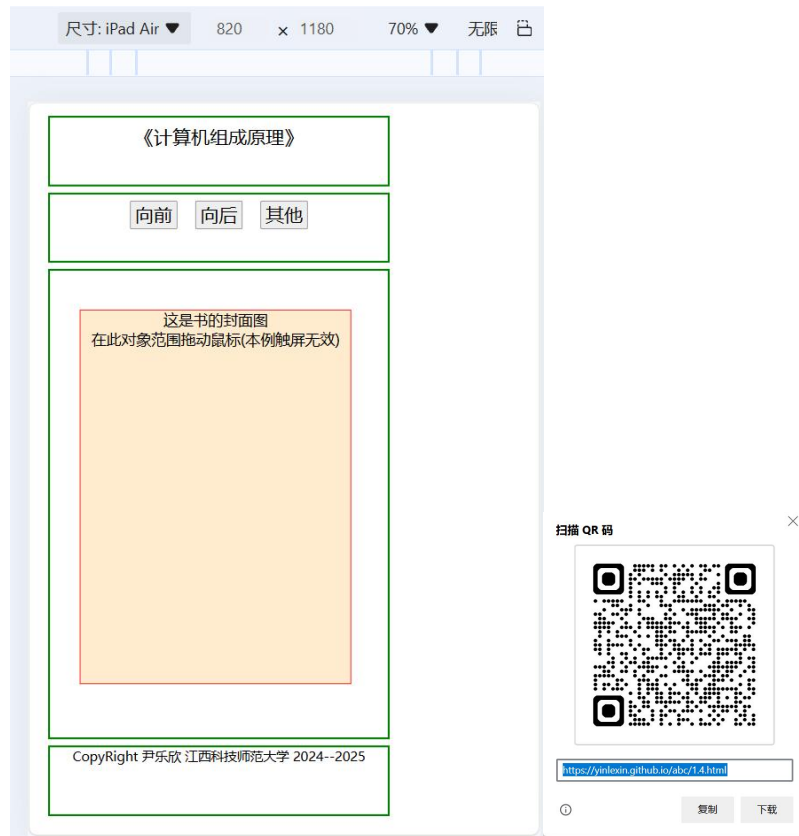


图 6.1 运行效果图

7.个性化 UI 设计中鼠标模型以及通用 UI 设计

7.1 分析和设计

在当今的软件与网页设计领域，个性化用户界面设计扮演着至关重要的核心角色。它不仅仅是简单的外观设计，更是对产品独特性和品牌形象的展现。通过个性化用户界面设计，产品能够更好地与用户进行情感连接，提升用户体验，增强产品的可访问性和包容性。此外，个性化设计还有助于促进产品的创新和发展，激发用户对产品的兴趣和忠诚度。在当今激烈的市场竞争中，重视个性化用户界面设计显得尤为必要。只有通过深度了解用户需求和习惯，精心打造个性化用户界面，才能真正实现产品与用户的良好互动，提升产品的市场竞争力。因此，对于设计和开发团队来说，将个性化用户界面设计纳入产品设计和开发的重要考量

之中，将为产品赢得更广泛的认可和青睐。

7.2 项目的实现和编程

根据之前的代码，增添了用户友好的界面设计。代码段中定义了一个名为 `Pointer` 的对象，用于处理鼠标和触摸屏之间的交互事件。在初始化阶段，`Pointer` 对象包括三个属性：`isDown`，用于检测鼠标或触摸屏是否按下；`x`，记录上次交互的 `x` 坐标；`deltaX`，用于计算滑动和拖动的位移量。基于这些属性，定义了三个事件处理功能：开始处理、结束处理和移动处理。当鼠标按下或触摸屏时，会激活 `handleBegin` 功能，将 `Pointer.isDown` 属性设为 `true`，并记录对应事件类型的坐标。然后更新特定 ID（"bookface"）的网页元素的文字，以反映活动类型和坐标。当鼠标松开或触摸屏时，执行相应功能，将 `Pointer.isDown` 属性设为 `false`，并阻止默认事件。基于屏幕事件和滑动距离，更新了"bookface"的文字和样式。此功能用于控制鼠标移动或触摸屏滑动。它首先拦截默认事件，然后检查屏幕事件，当 `Pointer.isDown` 为 `true` 时，计算并更新"bookface"元素的文字和样式。在鼠标事件中执行类似的逻辑。最后，通过自定义的 `$` 函数，将事件监听器绑定到网页上的元素（假设 ID 为"bookface"），以处理鼠标和触摸屏的交互操作。这些事件监听器包括鼠标点击、触摸、释放、结束、退出以及鼠标和屏幕的移动。当这些事件被触发时，相应的处理功能会被调用，如 `handleBegin`、`handleEnd`、`handleMoving`，以提供灵活的用户交互反馈。同时，也将键盘事件监听器绑定到整个主体元素。当用户按下网页上的键时，将触发一个匿名功能，将键的字符值附加到具有 ID"aid"的单元的文字内容上，实现简单的键盘输入。

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

```
</div>
```

```
<script>
```

```
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';
```

```
//尝试对鼠标和触屏设计一套代码实现 UI 控制
```

```
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 ;
});
} //Code Block    end

function $(ele){
    if (typeof ele !== 'string'){
        throw("自定义的$函数参数的数据类型错误，实参必须是字符串！");

        return
    }
    let dom = document.getElementById(ele) ;
    if(dom){
        return dom ;
    }else{
        dom = document.querySelector(ele) ;
        if (dom) {
            return dom ;
        }else{
            throw("执行$函数未能在页面上获取任何元素，请自查问题！");

            return ;
        }
    }
} //end of $

```

</script>

7.3 项目的运行与测试

项目的运行结果如图 7.1 所示：



图 7.1 运行效果图

8.UI 的个性化键盘控制

8.1 分析和设计

在这一步中，我使用 `keydown` 和 `keyup` 键盘的基本事件，以便为将来的键盘增加更多的空间。程式码段会先设定两个侦听程式来针对主体元素，以处理按键与释放键的操作。任何一个按键在键盘上被点击，`keydown` 事件就会被触发，并且一个适当的回调函数会被执行。在这个回调函数中，第一次调用(`ev.preventDefault`)来终止缺省事件，接着获取关键字的键值 (`ev.keyCode`) 和一个字符编码 (`ev.charCode`)。然后，使用一个定制的\$功能来找到这个网页中的一个元素，并且更新它的文字内容来显示这个按键和它的字符编码。

8.2 项目的实现和编程

//提出问题：研究利用"keydown"和"keyup"2 个底层事件，实现同时输出按键状态和文本内容

```
$("#body").addEventListener("keydown",function(ev){  
    ev.preventDefault();
```

```

    let k = ev.key;
    let c = ev.keyCode;
    $("#keyboard").textContent = "您已按键 : " + k + " , "+" 字符编码 : " + c;
  });
  $("body").addEventListener("keyup",function(ev){
    ev.preventDefault() ;
    let key = ev.key;
    let code = ev.keyCode;
    $("#keyboard").textContent = "松开按键 : " + key + " , "+" 字符编码 : " +
code;

    if (printLetter(key)){
      $("#typeText").textContent += key ;
    }
    function printLetter(k){
if (k.length > 1){ //学生必须研究这个逻辑的作用
      return false ;
    }
    let puncs = ['~',' ','!','@','#','$','%','^','&','*','(',')','-', '_','+', '=',,;,:', '<','>','?', '/', ' '];
    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)
});
} //Code Block End
function $(ele){
    if (typeof ele !== 'string'){
        throw("自定义的$函数参数的数据类型错误，实参必须是字符串！");
        return
    }
    let dom = document.getElementById(ele);
    if(dom){
        return dom;
    }else{
        dom = document.querySelector(ele);
        if (dom) {
            return dom;
        }else{
            throw("执行$函数未能在页面上获取任何元素，请自查问题！");
            return;
        }
    }
} //end of $

```

8.3 项目的运行与测试

项目的运行结果如图 8.1 所示：



图 8.1 运行效果图

9.本项目中的高质量代码

本文介绍了一种基于 MVC（Model-View-Controller）的设计方法，以明确划分软件逻辑、数据和用户接口。首先，建立 **PointerModel**，监控指针（鼠标或虚拟手指）的位置和状态（按下、移动、释放），并提供信息更新接口。然后设计 **PointerView**，根据模型数据和图形逻辑，在屏幕上显示指针。最后，设计 **PointerController**，侦测鼠标和触摸屏事件，并更新模型中的指针。在整合 MVC 时，创建和初始化模型、视图和控制器的实例。将控制器的事件监听程序与鼠标和触摸屏事件结合，确保模型和视图得到适当更新。控制器获取位置、状态等信息，并更新模型。模型状态变化时，触发事件并传递新状态给已登记的事件侦听程序，包括视图的侦听器。视图接收事件后，基于新数据更新显示。采用 MVC 设计模式，解耦软件，增强可维护性和扩展性，简化系统测试与调试。添加新输入装置或特性时，修改或扩充控制器或模型，无需改变视图，减少代码复杂度和耦合度。

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

Bash（Bourne Again SHell）是一种流行的 Unix shell 和命令语言，常用于在 Linux 和其他类 Unix 操作系统上进行脚本编写和命令行交互。它提供了丰富的工具和功能，使其成为许多系统管理员和开发人员的首选工具之一。Bash 具有以下几个经典工具和特性：

1.命令解释器：Bash 作为一个命令解释器，能够解释和执行用户输入的命令，包括系统命令、应用程序和自定义脚本。

2.脚本编写：Bash 脚本是一种文本文件，其中包含了一系列 Bash 命令，可以用来完成特定的任务，如自动化系统管理任务、批处理处理文件等。

3.通配符和通配符扩展：Bash 支持通配符（如*和?）用于匹配文件名，以及通配符扩展，允许使用花括号展开多个值。

4.环境变量和别名：Bash 允许用户定义和管理环境变量，以及创建和使用命令别名，从而简化命令行操作。

5.管道和重定向：Bash 支持管道操作（|）和重定向操作符（>、<、>>等），使得用户可以方便地将命令的输出作为另一个命令的输入，或者将输出重定向到文件中。

6.条件语句和循环：Bash 提供了条件语句（if-else）和循环（for、while）等控制结构，使得用户能够编写复杂的脚本来处理不同的情况和大量的数据。

7. 函数和参数传递：Bash 支持函数定义和调用，以及参数的传递，这使得用户能够编写可重用的代码块，并向脚本传递参数。

总的来说，Bash 作为一个强大的命令行工具和脚本语言，为用户提供了丰富的功能和灵活性，成为了 Unix 系统中不可或缺的一部分。

10.2 通过 github 平台访问项目

借助 github 网站的图形化上传文件界面将本项目中的代码上传至 github 后，可以进入 <https://yinlexin.github.io> 地址来访问。



图 10.1 项目实现图

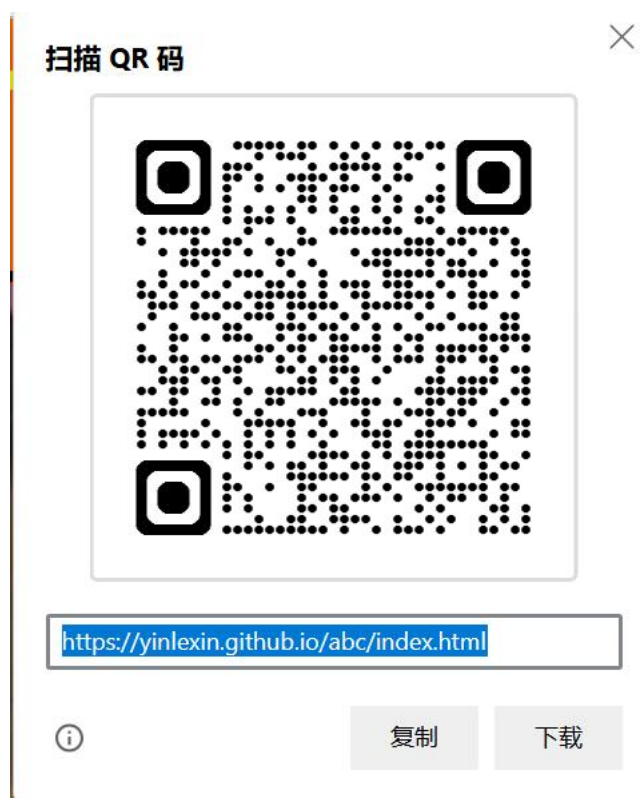


图 10.2 项目二维码

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