

第二次作业报告

刘叙杨 学号: 2023010838

liuxuyan23@mails.tsinghua.edu.cn

1 实现思路

本次作业包含登录页面和登录成功页面两个主要功能模块。

在页面外观上,在登录页面(index.html)中,设计了包含用户名和密码输入框、链接按钮、附加功能标签等元素在内的完整界面。登录成功页面(hw1/index_success.html)大体沿用了上一次作业的内容。

在技术实现方面,主要是使用了 javascript 完成主要的逻辑功能。

登录页面逻辑实现:

- **输入框焦点事件:** 通过 `addEventListener('focus')` 监听用户名输入框的焦点事件,当用户点击输入框时,通过 `classList.add('hidden')` 隐藏提示文字。
- **表单验证:** 在登录按钮点击事件中,通过 `querySelector` 获取用户名和密码输入框的值,检查是否为空。如果验证通过,使用 `localStorage.setItem('currentUsername', username)` 存储用户名,并通过 `window.location.href` 跳转到成功页面。
- **网络类型选择:** 通过 `querySelector('input[name="domain"]:checked')` 获取选中的网络类型复选框的值。

登录成功页面逻辑实现:

- **用户名显示:** 页面加载时通过 `DOMContentLoaded` 事件监听器,使用 `localStorage.getItem('currentUsername')` 获取存储的用户名,并更新到页面显示。
- **进度条动态更新:** 通过 `updateProgressBar()` 函数实现,解析流量使用文本(如"11.20 GB"),计算使用百分比(最大 50GB),并通过 `progressFill.style.width` 动态设置进度条宽度。
- **登出功能:** 监听登出按钮点击事件,使用 `localStorage.removeItem('currentUsername')` 清除存储的用户信息,并通过 `window.location.href = '../index.html'` 返回登录页面。

数据传递机制: 使用浏览器的 `localStorage` API 在两个页面间传递用户名信息,确保登录状态的持久性和页面间的数据一致性。

2 使用说明

将文件下载后,可以发现/src/index.html。可以直接用浏览器打开。此外/src/csstable 为样式表,/src/hw1 为上次作业所做页面,这里对于部分内容做了更改后拿来使用,src/source 为图片资源。

3 问题及解决办法

在实现过程中遇到的主要问题包括：

- **页面间数据传递**：最初考虑使用 URL 参数传递用户名，但发现 `localStorage` 更适合这种场景，能够保持数据的持久性。
- **绝对位置和相对位置的选择**：最初考虑使用绝对参数，后来在更换显示屏时发现这样做会导致不同显示屏上页面内容在不同位置，后改为使用相对位置。

4 参考资料

除了上次作业中的参考内容之外，本次作业中，为了实现和原网页样式高度一致，参考了网页源代码，使用了 `custom.css` 的部分参数，并使用了原网页的图片素材。此外使用了大模型进行注释的补全，以及代码格式的规范。