

# 作业二：假的校园网也能登陆

李晨宇 2020011366 lichenyu20@mails.tsinghua.edu.cn

## 实现思路

### 前端页面实现

对于前端的登陆界面，此次实现时将其划分为如图的四部分：



图1 登陆界面结构示意图

- 第一部分：由三个id分别为network、qrcode、at-tsinghua的div组成，分别用于显示清华校园网logo、二维码信息以及At-Tsinghua的信息。这三个组件实现思路类似，只需要将背景图片设置为对应logo，设置为no-repeat，再调整其大小、位置关系即可。
- 第二部分：该部分在实现时，首先实现的是底层的橙色部分。橙色部分的文字部分由资源文件导入，然后调整其位置与大小即可。再实现登陆信息框部分。该部分主要由两部分组成，第一部分为form，主要负责用户名、密码提示信息的显示以及输入框的显示，并且由于后续向后端发送用户输入的信息；第二部分为提交按钮，负责页面的跳转以及向后端发送信息。
- 第三部分：系统信息和组件部分实现思路均为：导入资源图片作为背景->写入对应文字->调整布局，此处不再赘述。
- 第四部分：该部分主要对文字的样式进行设置，此处不再赘述。

另外，值得一提的是，在本次前端界面的实现过程中，考虑到移动端兼容性的问题，进行响应式处理，在屏幕较小（不超过768px）的情形中，会对页面的关键信息和输入框进行放大，对用户较为友好。具体效果如下。



图2 不进行响应式处理以及进行响应式处理效果对比图(左侧为不进行响应式处理的效果，右侧为进行响应式处理的效果)

## 页面跳转以及前端向后端发送信息

### 登陆页面->连接界面

在实现该部分时，将第一个文本框以及第二个输入框中的信息分别与 `userID` 和 `userPw` 变量进行绑定，并将连接按钮按下的事件与 `login` 函数进行绑定。当用户在登录界面中按下连接按钮时，会触发 `login` 函数。`login` 函数首先会判断两个输入框中是否有内容，若无内容会弹窗提醒，如图3。



图3 连接按钮弹窗示意图

若输入是合法的，则 login 函数会用post方法与后端进行发送，将用户名 userID 发送给后端，并将 userID 以及后端传回的使用流量信息 usage\_value 一并传给内容页面，并通过路由跳转至第二个页面。

连接页面->登录页面

在连接页面中，根据登录页面中传来的信息，将其与控制用户名显示、流量信息显示的组件进行绑定，从而实现动态显示流量信息的效果。再将id为disconnect的按键的按下事件与函数 disconnect 进行绑定。disconnect 函数利用post方法，将当前用户的 userID 传给后端，后端会给这个 userID 对应的用户加上固定流量。

后端部分

后端部分中，主要实现如下两个函数：

| 函数名        | 函数功能         |
|------------|--------------|
| login      | 获取某用户已使用流量信息 |
| disconnect | 保存某用户已使用流量信息 |

## login 函数

该函数获取从前端发送的信息，将其与字典 `user_dict` 中的信息进行比对。若该用户为第一次连接，则将其流量设置为0并返回给前端；若非第一次连接，则将原来使用的流量信息返回。

```
1 @app.route('/login', methods=['POST', 'GET'])
2 def login(): # login函数, 可根据传进来的用户名获取对应的流量信息
3     if request.method == 'POST':
4         userID = request.get_data(as_text = True)
5         if userID in user_dict:
6             pass
7         else:
8             user_dict[userID] = '0'
9         return user_dict
10    else:
11        print('here\n')
12        return 'here\n'
```

## disconnect 函数

该函数获取从前端发送的信息，将该用户对应的流量值加上固定数值（5G）然后对字典 `user_dict` 中对应的信息进行更新保存。

# 遇到的困难及解决办法

## 前端部分

本次前端选用了VUE-CLI。VUE-CLI对于新手不是很友好，刚开始看到复杂的目录以及各种配置文件感觉一筹莫展，甚至不知道页面要写在哪，因此在前端页面的实现过程中耗费了较多时间。为了解决该问题，我查阅了很多资料，其中包括但不限于各大论坛资料(如CSDN)、B站学习资源(跟着b站课程熟悉VUE-CLI并先通过一个小项目练手)、ACWING课程(在该课程中，我跟着老师实现了一个简易的博客网站，该网站支持关注、登陆等功能)。

## 后端部分

由于头一次接触后端以及flask框架，在刚开始实现后端的时候对于路由等机制不是很熟悉，在这部分上浪费了较多的时间。为此我查阅了菜鸟教程以及flask中文官方文档进行了学习。

## 使用说明

本次作业的代码位于 `src` 文件夹，其结构如下(略去modules文件夹部分)：

```
1 | .
2 | └─ README.md
3 | └─ babel.config.js
4 | └─ global.js
5 | └─ jsconfig.json
6 | └─ package-lock.json
```

```
7 |─ package.json
8 |─ public
9 |   │─ favicon.ico
10 |   └─ index.html
11 |─ server.py
12 |─ node_modules
13 |─ src
14 |   │─ App.vue
15 |   │─ assets
16 |       │─ android.gif
17 |       │─ at-tsinghua.png
18 |       │─ background-login.png
19 |       │─ bg.png
20 |       │─ bookmark.gif
21 |       │─ connect.png
22 |       │─ contact.gif
23 |       │─ content_shadow.png
24 |       │─ disconnect_bar.png
25 |       │─ download.png
26 |       │─ help.gif
27 |       │─ hi.png
28 |       │─ info_logo.gif
29 |       │─ ios.gif
30 |       │─ learn_logo.gif
31 |       │─ lib_logo.gif
32 |       │─ linux.gif
33 |       │─ macos.gif
34 |       │─ mail_logo.gif
35 |       │─ network_logo.png
36 |       │─ notification.png
37 |       │─ qrcode.jpeg
38 |       │─ usege_bar.gif
39 |       │─ welcome.png
40 |       │─ wifi_logo.gif
41 |       └─ windows.gif
42 |   │─ components
43 |   │─ main.js
44 |   │─ router
45 |       │─ index.js
46 |   │─ store
47 |       │─ index.js
48 |   └─ views
49 |       │─ ConnectView.vue
50 |       └─ HomeView.vue
51 └─ vue.config.js
```

运行代码时，实现先用vue-cli工具运行前端界面，然后在用 `python3 ./src/server.py` 命令运行后端。

代码运行视频链接：<https://cloud.tsinghua.edu.cn/f/aeef7757493947a7ba3b/>

