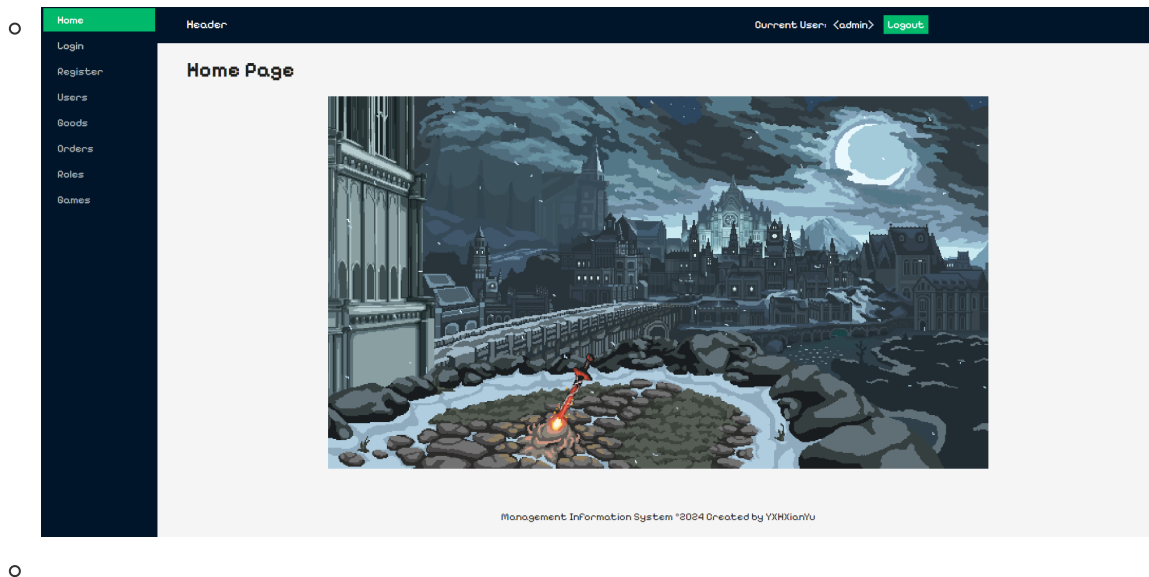


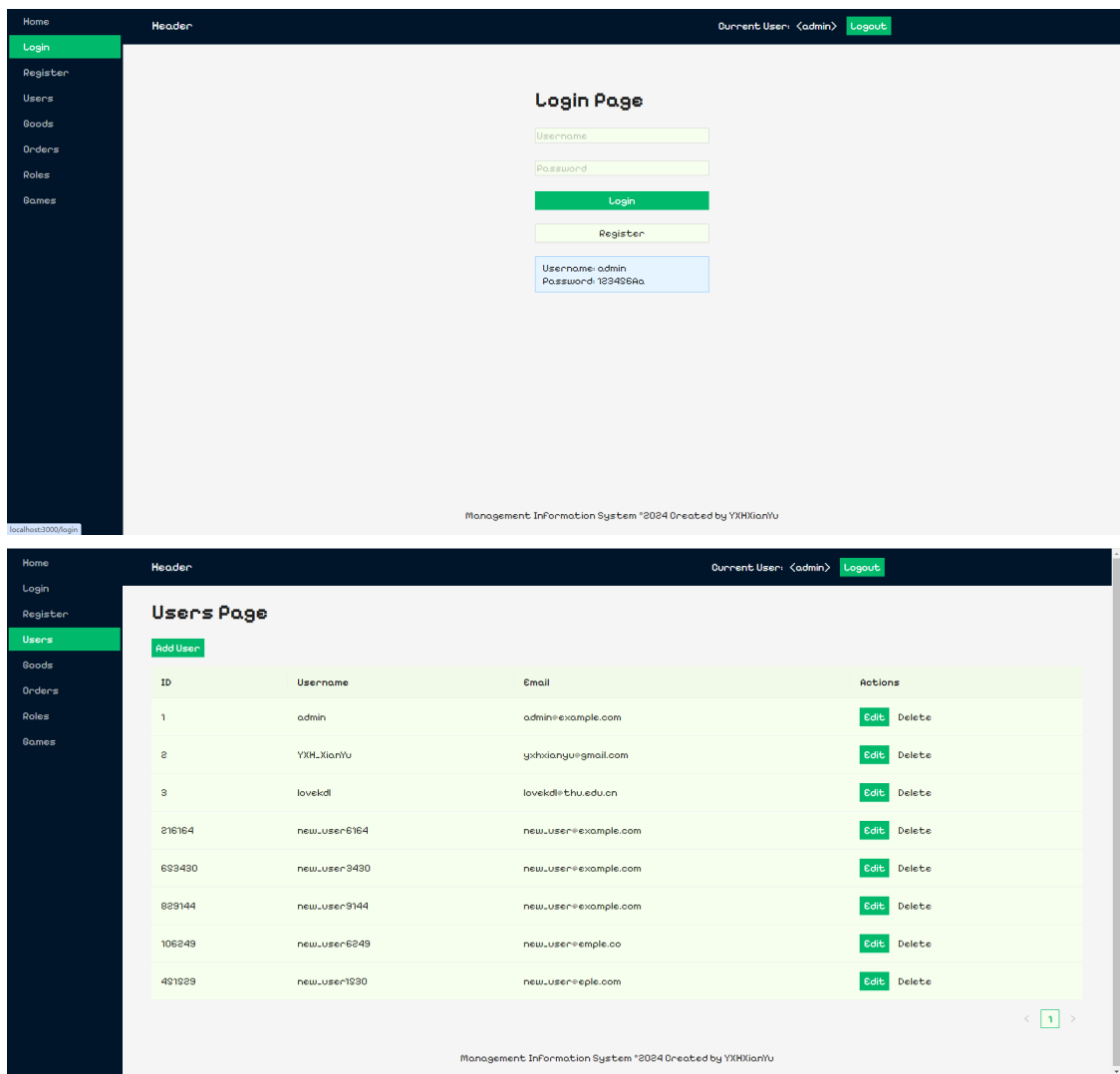
轻量化第三次小作业

21301114 俞贤皓

1. 简介

- 作业内容：信息管理系统
- 本次作业内容开源在 [我的Github](#)
- 我完成了以下功能
 - 纯React静态页面 60分
 - 后台用户管理 10分
 - 数据持久化 10分
 - 后端联动 20分
 - 使用 node.js / express.js 实现（PPT说加分）
 - **总分 100分** (60+10+10+20)
- 关于模板参考
 - **我没有参考任何老师的模板**
 - 我在 **第一次** 完成本次作业的时候，参考了老师的模板。但在通知后，我 **完全从零开始，重新编写了本次作业**，并且 **特意以不同的架构实现本次作业**，目前和老师的模板没有任何相似之处。
- 预览





2. 代码说明

- 前端位于 `./frontend`
 - 构建: `npm install`
 - 运行: `npm start`
 - 基于 React + Ant Design
 - 预编译版本位于 `./frontend/build`

- 在 `./frontend` 目录下执行

```
npm install -g serve  
serve -s ./build
```

- 即可启动前端

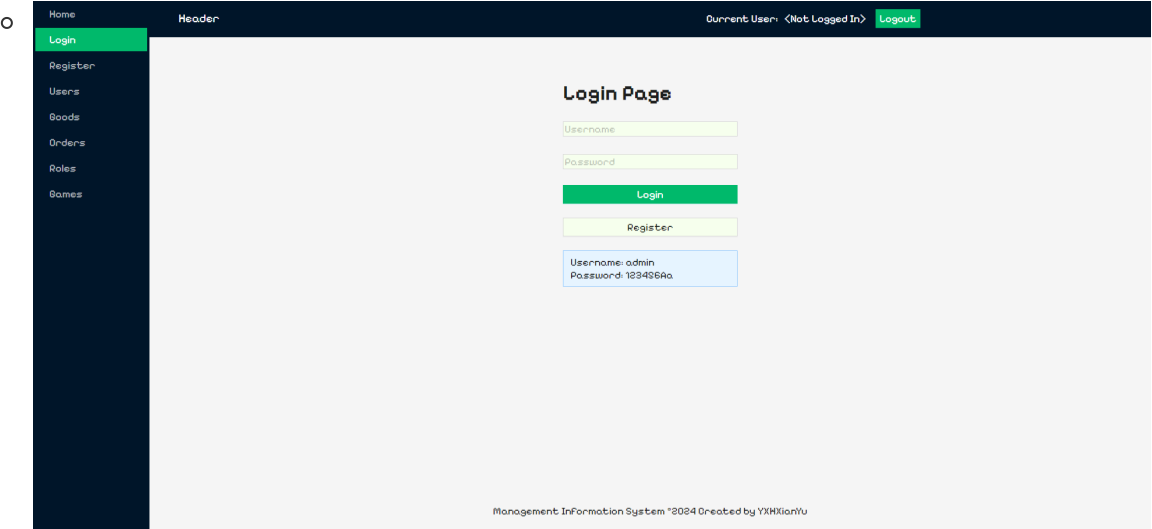
- 后端位于 `./backend`
 - 构建: `npm install`
 - 运行: `npm start` 或 `node ./index.js`
 - 使用 **express.js** 实现的一个极端轻量化的Redis风格后端

- 后端无预编译版本（因为本身就只有单文件）

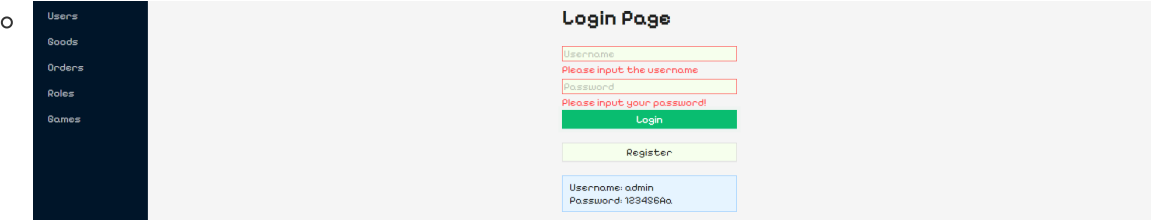
3. 功能说明

3.1 纯React静态页面（60分）

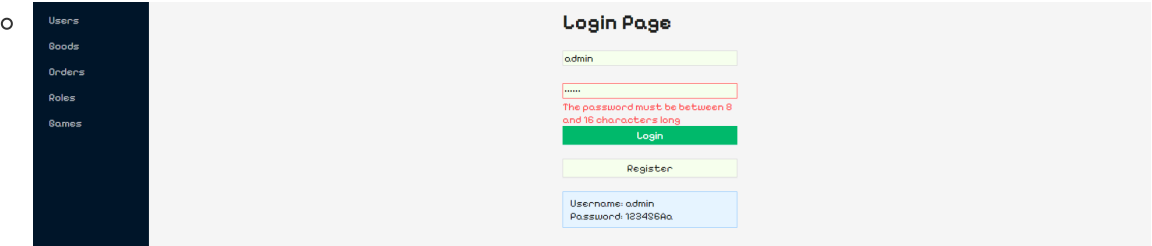
- 登录页面



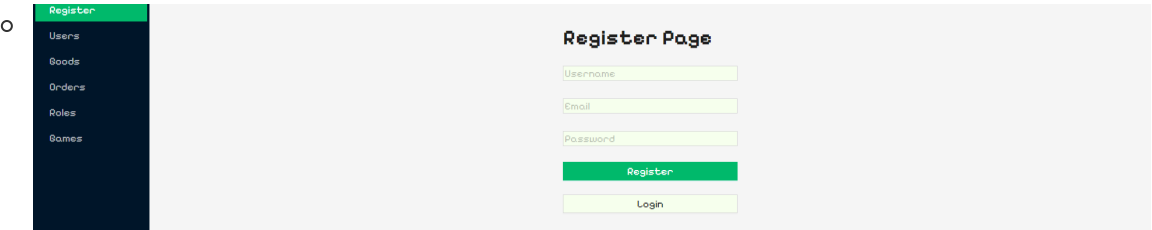
- 登陆页面（缺少输入）



- 登陆页面（密码不够长）

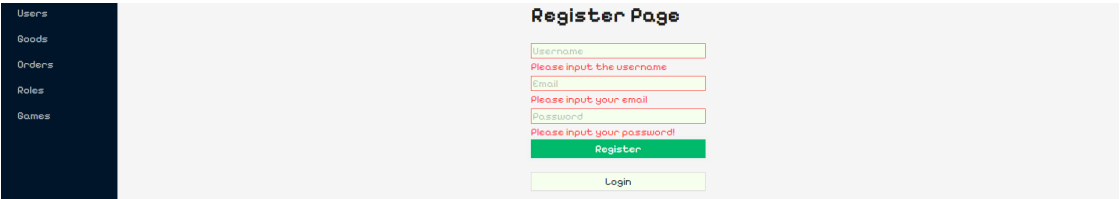


- 注册页面

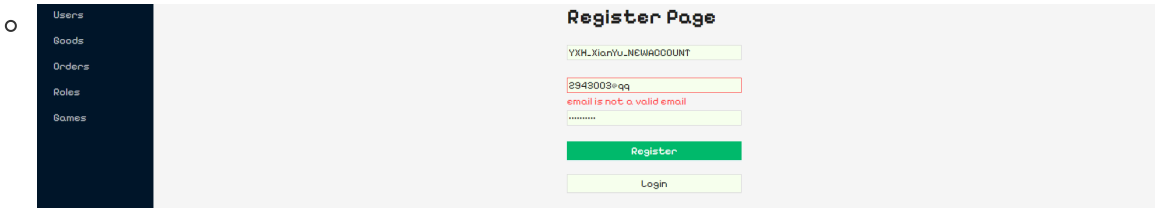


- 注册页面（缺少输入）

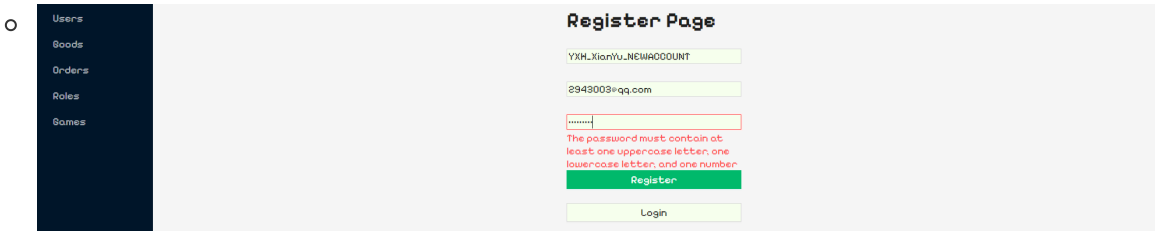
◦



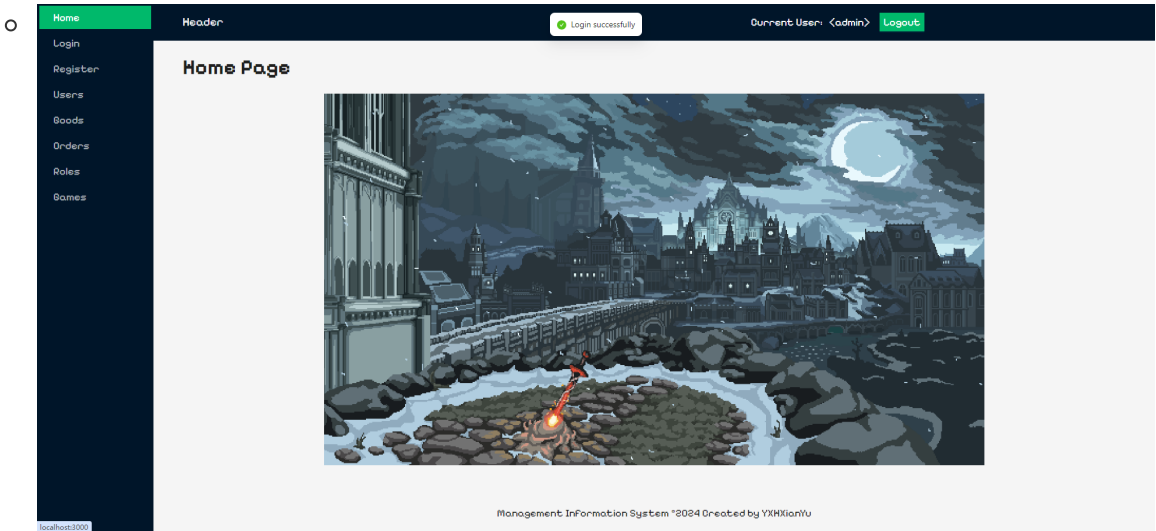
- 注册页面（邮箱格式不正确）



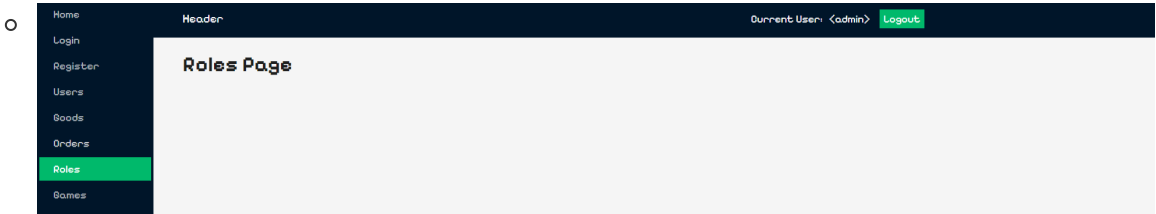
- 注册页面（密码缺少大写字符或小写字符）



- 主页面

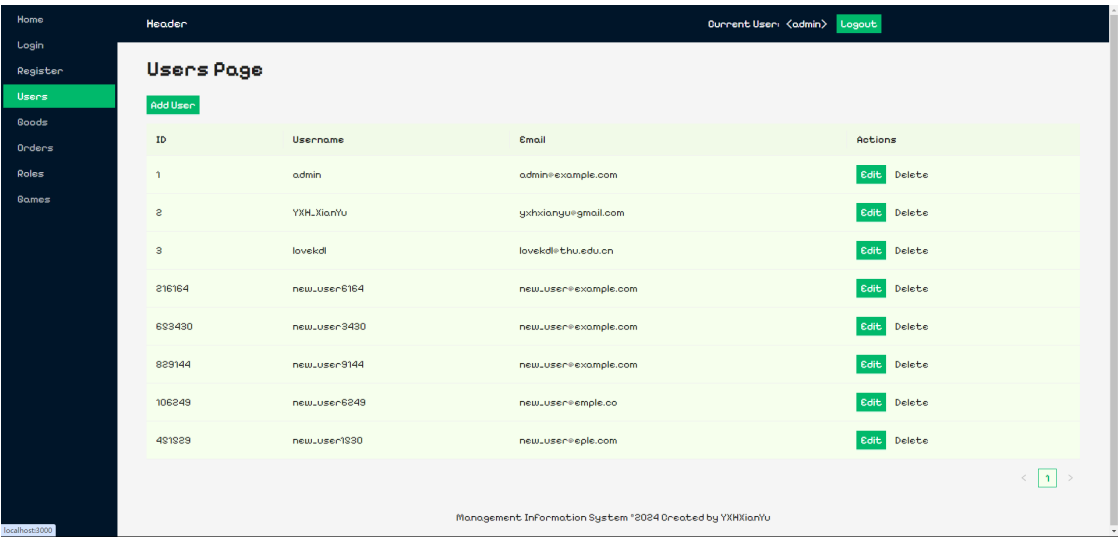


- 其他页面



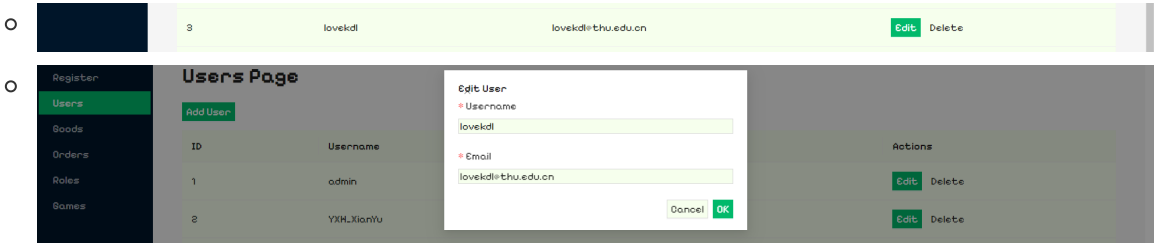
- 用户管理界面

-



3.2 后台用户管理（10分）

- 修改用户（修改lovekdl用户的邮箱，修改前）



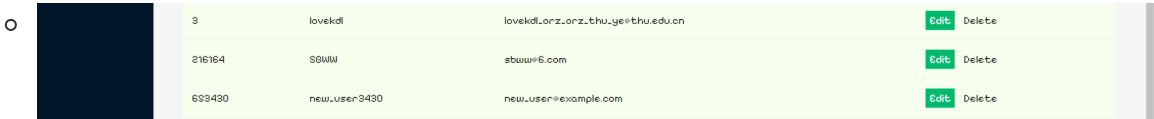
- 修改用户（修改lovekdl用户的邮箱，修改后）



- 随机添加用户（添加到了第二页）



- 删除用户（删除前）



- 删除用户（删除后）



- 重启浏览器，数据修改仍存在



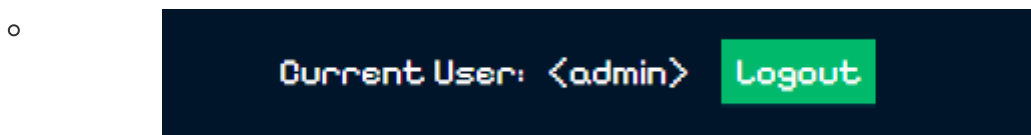
Orders				
Roles				
Games				
	1	admin	admin@example.com	Edit Delete
	2	YXH_XianYu	yxhxiangyu@gmail.com	Edit Delete
	3	lovekd1	lovekd1orz.orz.thu.ye@thu.edu.cn	Edit Delete
	683430	new.user3430	new.user@example.com	Edit Delete
	829144	new.user9144	new.user@example.com	Edit Delete
	106249	new.user6249	new.user@emple.co	Edit Delete

6.3 数据持久化（10分）

- 经检查，数据已经同步到后端数据库中

```
package.json 16 {
  / prpm-lock.yml 17   "id": 2,
                                18   "username": "YXH_XianYu",
                                19   "password": "1008453",
                                20   "email": "yxhxiangyu@gmail.com"
                                21 }
                                22
                                23   "id": 3,
                                24   "username": "lovekd1",
                                25   "password": "1008453",
                                26   "email": "lovekd1orz.orz.thu.ye@thu.edu.cn"
                                27 }
                                28
                                29   "id": 683430,
                                30   "username": "new.user3430",
                                31   "password": "password",
                                32   "email": "new.user@example.com",
                                33 }
                                34
                                35   "id": 829144,
                                36   "username": "new.user9144",
                                37   "password": "password",
                                38   "email": "new.user@example.com"
                                39 }
```

- 同时，重启浏览器或者刷新网页，都不会取消登陆状态（需要手动点击Logout退出登陆状态）



6.4 后端联动（20分）

- 使用express.js实现
- 后端日志

```
2024/5/27 14:28:09 get { key: 'userList' }
true
2024/5/27 14:28:18 set userList
2024/5/27 14:28:21 set userList
2024/5/27 14:28:24 get { key: 'userList' }
true
2024/5/27 14:28:24 get { key: 'userList' }
true
2024/5/27 14:28:24 get { key: 'userList' }
true
2024/5/27 14:28:24 get { key: 'userList' }
true
2024/5/27 14:28:24 get { key: 'userList' }
true
2024/5/27 14:28:24 get { key: 'userList' }
true
2024/5/27 14:28:27 get { key: 'userList' }
true
2024/5/27 14:28:57 set userList
```

- 后端核心代码

```
11 app.post('/set', (req, res) => {
10   console.log(getTime(), chalk.red('set'), req.body.key)
9   const { key, value } = req.body
8   if (!key || value === undefined) {
7     return res.status(400).json({ error: 'Key and value are required' })
6   }
5   const data = readData()
4   data[key] = value
3   writeData(data)
2   res.json({ success: true })
1 }

44 You, 12 hours ago · Backend
1 app.get('/get/:key', (req, res) => {
2   console.log(getTime(), chalk.red('get'), req.params)
3   const { key } = req.params
4   const data = readData()
5   console.log(key in data)
6   if (!(key in data)) {
7     return res.status(404).json({ error: 'Key not found' })
8   }
9   // console.log(chalk.red('get result:'), data[key])
10  res.json({ key, value: data[key] })
11 }
```

- 前端交互部分代码：

○

```

2 export const load = async (key) => {
1   try {
14     const response = await axios.get('http://localhost:5000/get/' + key);
1     if (response.status === 200) {
2       return response.data.value;
3     } else {
4       console.log('Error getting data (Response status is not 200): ', response.status);
5       return null;
6     }
7   } catch (error) {
8     console.log('Error getting data: ', error);
9     return null;
10  }
11 }

```

- ```

12
13 export const save = async (key, value) => {
14 const data = {
15 'key': key,
16 'value': value,
17 }
18 try {
19 await axios.post('http://localhost:5000/set', data);
20 return true;
21 } catch (error) {
22 console.log('Error setting data: ', error);
23 return null;
24 }
25 }

```

- 前端异步代码：初始化 & 获取用户数据

- ```

31 // Initialize user data
30 useEffect(() => {
29   const func = async () => {
28     const user = await load('user')
27     const cur_user_id = localStorage.getItem('cur_user_id')
26     if (user) {
25       setData({
24         ...data,
23         user: user,
22         cur_user_id: cur_user_id,
21       })
20     } else {
19       message.error('Backend server is not running or not connected to the Internet')
18       setData({
17         ...data,
16         user: null, // Backend is down
15         cur_user_id: cur_user_id,
14       })
13     }
12   }
11   func()
10   // eslint-disable-next-line
9 }, [])

```

- 前端异步代码：刷新Header的Current User

- ```

7 // Update current username
6 useEffect(() => {
5 const func = async () => {
4 console.log(data)
3 let cur_username
2 if (data.user === null || data.user === undefined) {
1 } else if (data.cur_user_id === undefined || data.cur_user_id === 'undefined') {
43 cur_username = 'Not Logged In'
1 } else {
2 for (let i = 0; i < data.user.length; i++) {
3 // eslint-disable-next-line
4 if (data.user[i].id == data.cur_user_id) {
5 cur_username = data.user[i].username
6 break
7 }
8 }
9 }
10 setData({
11 ...data,
12 cur_username: '<' + cur_username + '>',
13 })
14 }
15 func()
16 // eslint-disable-next-line
17 }, [data.user, data.cur_user_id])
18

```

< 演示结束 >

#### 4. 关于我第一次实现的本次作业

- 因为我这周末要参加南大的夏令营，复习时间很紧。第一次实现作业时，认真写了css，但第二次实现时就没有写css了🤔。所以在这里纪念一下我第一次实现本次作业的UI和界面（笑）。

