Web前端开发-作业5

1. 运行说明

1. 安装 Node.js：终端输入 `npm install` ；
2. 输入`node server.js` ，启动服务端；
3. 在 WebStorm 或浏览器中打开 `login.html` 以进入登录页面。

2. 实验功能和设计

1. 使用 `node.js` 框架；
2. 使用本地 MySQL 数据库；
3. 可以使用邮箱注册/登录完成整体流程，第三方账号体现在登录界面；
4. 注册页面应使用正则表达式提示密码强度；
5. 登录页面可与一级页面结合，登陆后自动跳转到一级页面；
6. 使用 HTTP 基本鉴权。

3. 数据库配置

使用本地 MySQL 数据库。

数据结构和初始化如下：

```
CREATE DATABASE IF NOT EXISTS WebProgramming CHARACTER SET UTF8;
USE WebProgramming;
DROP TABLE IF EXISTS `user`;
CREATE TABLE `user` (
    `email` varchar(64) NOT NULL,
    `name` varchar(64) NOT NULL,
    `password` varchar(64) NOT NULL,
    PRIMARY KEY (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

4. 注册

只有在登陆页面的邮箱和密码输入框为空时，才能够点击注册按钮进入注册页面。

4.1 用户名

不同邮箱注册的用户用户名允许重复。

4.2 密码

4.2.1 实现方案

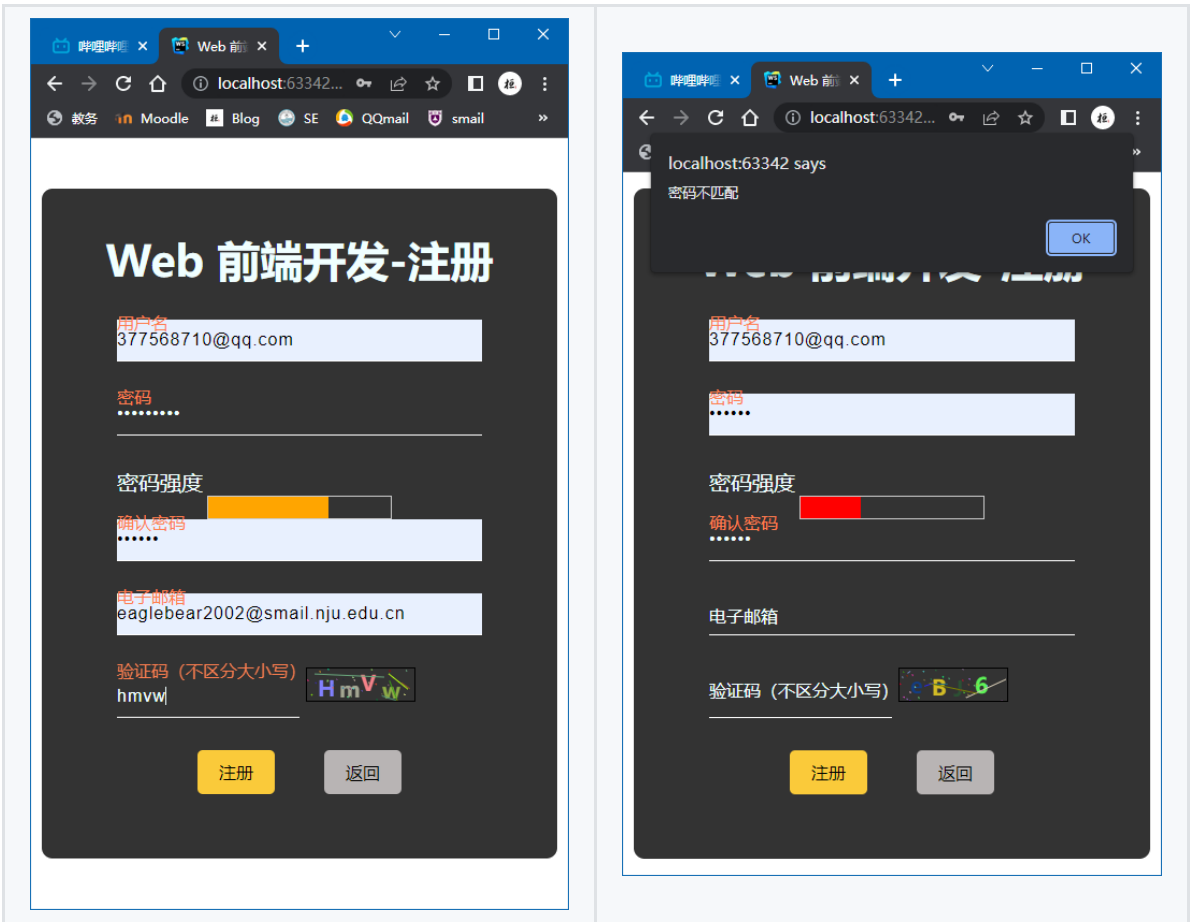
笔者参考 浅谈密码强度规则的5个版本 中的常规版密码强度规则，实现了自己的密码强度规则：

1. 初始密码级别为 0；
2. 当密码包含小写字母时，密码级别加 1；
3. 当密码包含大写字母时，密码级别加 1；
4. 当密码包含数字时，密码级别加 1；
5. 当密码包含字母和数字以外的特殊字符时，密码级别加 1；

6. 当密码长度小于 6 时，密码级别为 0；
7. 密码强度最高为 3；
8. 密码级别为 0、1、2、3 分别对应不满足条件、弱、中、强。

4.2.2 实现功能

1. 前端计算用户注册使用的的密码的密码强度；
2. 不满足其条件的密码无法通过注册；
3. 在注册过程中，密码强度等级使用不同颜色和长度的矩形来表示；
4. 如果两次输入的密码不同，网页提示：“密码不匹配”。



4.3 电子邮箱注册

1. 输入的电子邮箱必须是没有注册过的，否则会收到网页提示：“该邮箱已被注册”。

4.4 验证码

4.4.1 实现方案

笔者在前端使用 canvas 直接生成验证码，不使用后端接口，这种方法有有如下优点：

1. 减少后端接口的使用，减轻服务器负担；
2. 前端直接生成验证码，降低生成传输验证码的前后端通信开销。

这种方法还有如下缺点：

1. 用户可能会使用技术手段绕过前端生成的验证码，给系统造成安全风险。

4.4.2 实现功能

1. 用户需要输入正确的验证码才能完成注册流程；
2. 点击验证码图片即可更换验证码；
3. 输入验证码错误会收到网页提示： “验证码错误”

4.5 注册成功

成功注册账号，页面提示：“注册成功”并自动跳转至登陆页面。

5. 登录

1. **一般的浏览器会自动填充被保存的用户名**，需要用户手动填入注册的**邮箱**；
2. 输入已经注册的邮箱地址和密码即可成功登录并自动跳转到一级页面；
3. 如果输入的邮箱未注册，网页提示：“邮箱不存在”；
4. 如果输入的密码与注册的密码不同，网页提示：“密码错误”。