

web 登录实验说明文档

实现内容

1. 通过 express 框架实现注册和登录功能，详见 loginServer.js
2. 通过 mongodb 存放用户的数据，详见 models.js；数据表 导出到 users.json
3. 通过 bcrypt 实现了密码加密，详见 models.js

```
// 在数据库中新建一个表，表中只有用户名和密码这两列
const Userscheme = new mongoose.Schema( definition: {
  username: { type: String , unique: true},//保证用户名唯一
  password: {
    type: String,
    set(val) {
      return require('bcryptjs').hashSync(val,4)//对密码加密
    }
  },
});
```

4. 通过 ajax 发送 post 请求，详见 login.html 和 register.html

```

// 使用ajax传输数据
$.ajax({
    type: "POST",
    dataType: "json",
    crossDomain: true, // 允许跨域请求
    url: 'http://localhost:3030/api/register',
    contentType: "application/json",
    // 传输的数据本体
    data: JSON.stringify({
        'username': username,
        'password': password,
    }),
    // 对返回后的结果进行判断
    success: function (result) {
        console.log(result);
        alert("注册成功");
        window.location.href = "login.html";
    },
    error: function (result) {
        console.log(result);
        alert("注册失败");
    }
})
}

```

5. 实现了验证码验证, 详见 login.html, 通过 createCode()函数生成随机的验证码, 在发送请求前验证验证码正确与否, 若验证码正确才发送 post 请求

```

// 登录函数
function login() {
    var username = document.getElementById("username").value;
    var password = document.getElementById("password").value;
    if (username === '') {
        alert("Username can't be empty!");
        return false;
    }
    if (password === '') {
        alert("Password can't be empty!");
        return false;
    }

    // 验证输入的验证码是否正确
    var icode = document.getElementById("icode").value.toUpperCase();
    if (icode !== code) {
        alert("Verification Code is wrong!");
        return false;
    }

    $.ajax({
        type: "POST",
        dataType: "json",
        crossDomain: true,
        url: 'http://localhost:3030/api/login',
        contentType: "application/json",
        data: JSON.stringify({
            'username': username,
            'password': password,
        }),
        success: function (result) {

```

6. 采用正则表达式规定了密码的格式，规定了用户名为 4-20 位的字母或数字，密码为 6-16 位的字母或数字

```

<form action="homePage.html"> <!-- 当提交表单时跳转到action里相对 URL - 指向网站内的一个文件-->
<div class="input-box">
    <input type="text" id="username" name="username" pattern="^[a-zA-Z0-9]{3,19}$" class="form-control" placeholder="请输入用户名">
</div>
<div class="input-box">
    <input type="password" id="password" name="password" pattern="^[a-zA-Z0-9]{3,19}$" class="form-control" placeholder="请输入密码">
</div>

```

7. 实现了简单的密码轻度提示，当密码为空时提示“password is blank”；当密码不足 6 位时提示“not long enough”；当密码为 6-10 位时提示“could be stronger”；当密码为 10-16 位时提示“strong enough”；当密码超过 16 位时提示“too long”。详见 login.html

8. 界面截图

