

XJCO2011 Web Application Development

Design Document

Coursework 2

Chenhao Zhao [SWJTU ID: 2019110033]

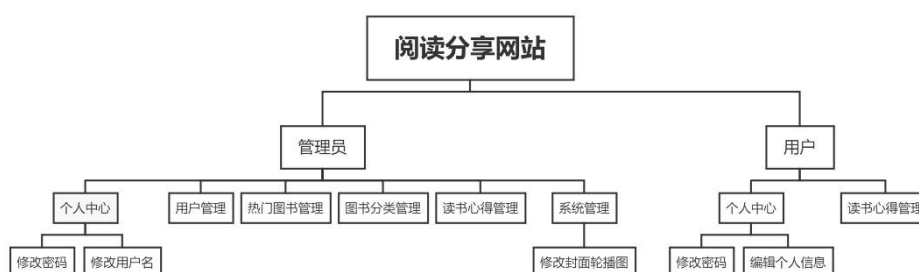
2021-12-8

Abstract

This design document is divided into three sections. The first part is the introduction of this project and its main functions which includes the purpose of the website and Functional module diagram. Then, the second part is the analysis of the website. After that, the third part is the self-evaluation.

1. Introduction of the project and man functions

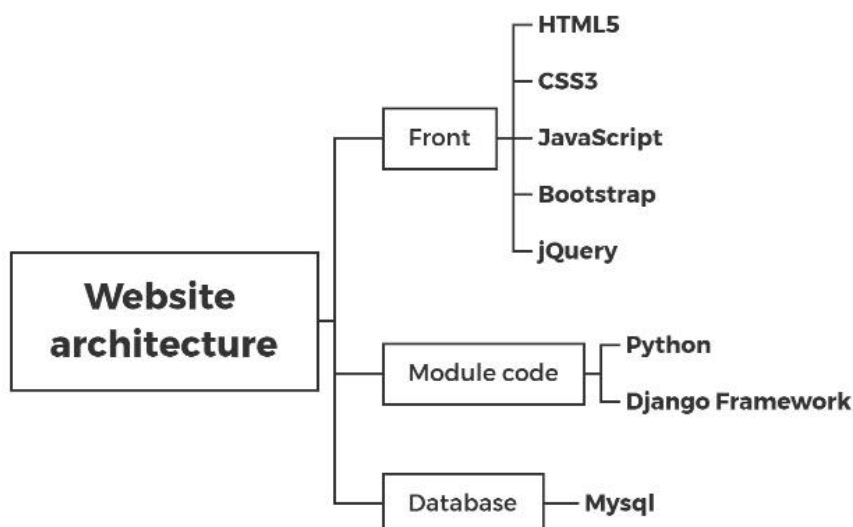
This project is a reading sharing website which built in Python. The front end of the web page, based on the Django structure, is built according to HTML5, CSS3 standards and the use of Bootstrap to style the website. The database implements data management through MySQL. The functional module diagram of this website is shown below.



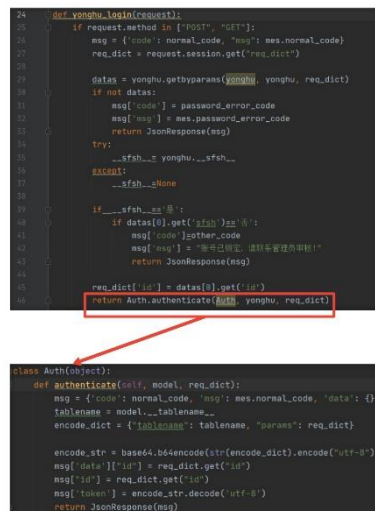
2. Analysis of the website

2.1 Web forms

- The following image shows the basic structure of the site.



- Client-side/server-side authentication has been implemented to ensure the integrity of the Web application.



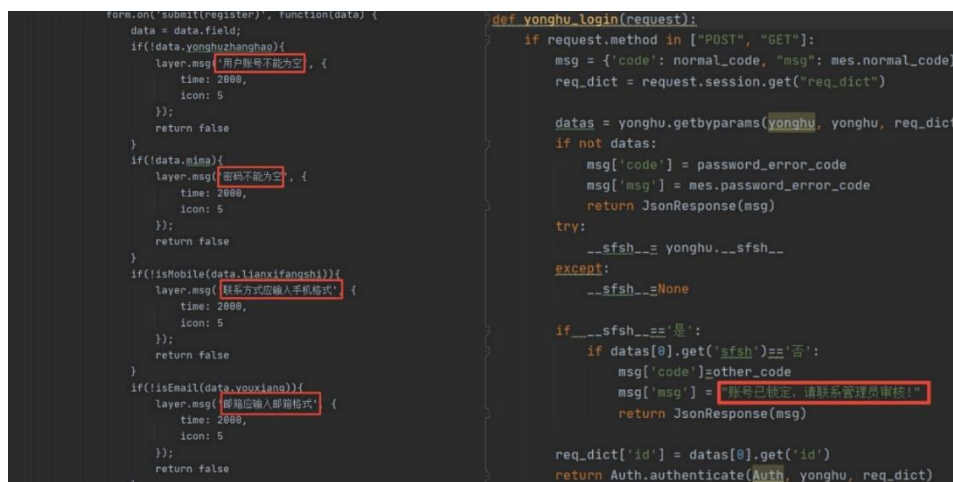
```

def yonghu_login(request):
    if request.method in ["POST", "GET"]:
        msg = {'code': normal_code, 'msg': mes.normal_code}
        req_dict = request.session.get("req_dict")
        datas = yonghu.getbyparams(yonghu, yonghu, req_dict)
        if not datas:
            msg['code'] = password_error_code
            msg['msg'] = mes.password_error_code
            return JsonResponse(msg)
        try:
            __sfsh__ = yonghu.__sfsh__
        except:
            __sfsh__ = None
        if __sfsh__ == '是':
            if datas[0].get('id') == '否':
                msg['code'] = other_code
                msg['msg'] = "账号已锁定, 请联系管理员审核!"
                return JsonResponse(msg)
            req_dict['id'] = datas[0].get('id')
            return Auth.authenticate(Auth, yonghu, req_dict)

class Auth(object):
    def authenticate(self, model, req_dict):
        msg = {'code': normal_code, 'msg': mes.normal_code, 'data': {}}
        tablename = model.__tablename__
        encode_dict = {'tablename': tablename, 'params': req_dict}
        encode_str = base64.b64encode(str(encode_dict).encode('utf-8'))
        msg['data']['id'] = req_dict.get('id')
        msg['id'] = req_dict.get('id')
        msg['token'] = encode_str.decode('utf-8')
        return JsonResponse(msg)

```

- At the same time, it provides information feedback to users when they make incorrect input.



```

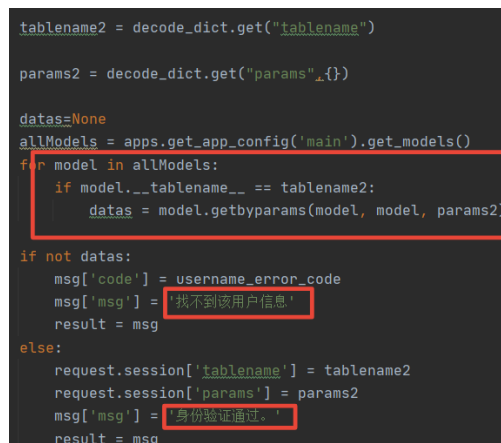
form.on('submit(register)', function(data) {
    data = data.field;
    if(!data.yonghuzhanghao){
        layer.msg("用户账号不能为空", {
            time: 2000,
            icon: 5
        });
        return false
    }
    if(!data.mima){
        layer.msg("密码不能为空", {
            time: 2000,
            icon: 5
        });
        return false
    }
    if(!isMobile(data.lianxifangshi)){
        layer.msg("联系方式应输入手机格式", {
            time: 2000,
            icon: 5
        });
        return false
    }
    if(!isEmail(data.youxian){
        layer.msg("邮箱应输入邮箱格式", {
            time: 2000,
            icon: 5
        });
        return false
    }
});

def yonghu_login(request):
    if request.method in ["POST", "GET"]:
        msg = {'code': normal_code, 'msg': mes.normal_code}
        req_dict = request.session.get("req_dict")
        datas = yonghu.getbyparams(yonghu, yonghu, req_dict)
        if not datas:
            msg['code'] = password_error_code
            msg['msg'] = mes.password_error_code
            return JsonResponse(msg)
        try:
            __sfsh__ = yonghu.__sfsh__
        except:
            __sfsh__ = None
        if __sfsh__ == '是':
            if datas[0].get('sfsh') == '否':
                msg['code'] = other_code
                msg['msg'] = "账号已锁定, 请联系管理员审核!"
                return JsonResponse(msg)
            req_dict['id'] = datas[0].get('id')
            return Auth.authenticate(Auth, yonghu, req_dict)

```

2.2 Database

- Client/server validation is applied to ensure that the data in the database is consistent and to provide feedback to the user when incorrect input is provided.



```

tablename2 = decode_dict.get("tablename")
params2 = decode_dict.get("params", {})
datas=None
allModels = apps.get_app_config('main').get_models()
for model in allModels:
    if model.__tablename__ == tablename2:
        datas = model.getbyparams(model, model, params2)
if not datas:
    msg['code'] = username_error_code
    msg['msg'] = "找不到该用户信息"
    result = msg
else:
    request.session['tablename'] = tablename2
    request.session['params'] = params2
    msg['msg'] = "身份验证通过."
    result = msg

```

- Many-to-many relationships are implemented. This can be simply understood as: a user can publish a reading impression on many books, and a book can own many reading impressions by multiple users.



2.3 Authentication & sessions

- After registering and logging in this website, users can publish their own reading experience. Additionally, users can also manage their own published reading experience, including adding, modifying, and deleting, to get a personalized experience.

The screenshot displays the 'Reading Sharing Website' (阅读分享网站) interface. The user is logged in as '用户 123' and can perform actions like '退出到前台' (Exit to Frontend) or '退出登录' (Exit Login).

The left sidebar contains navigation options: 首页 (Home), 个人中心 (Personal Center), 修改密码 (Change Password), 个人信息 (Personal Information), and 读书心得管理 (Reading Impression Management). The '读书心得管理' option is highlighted.

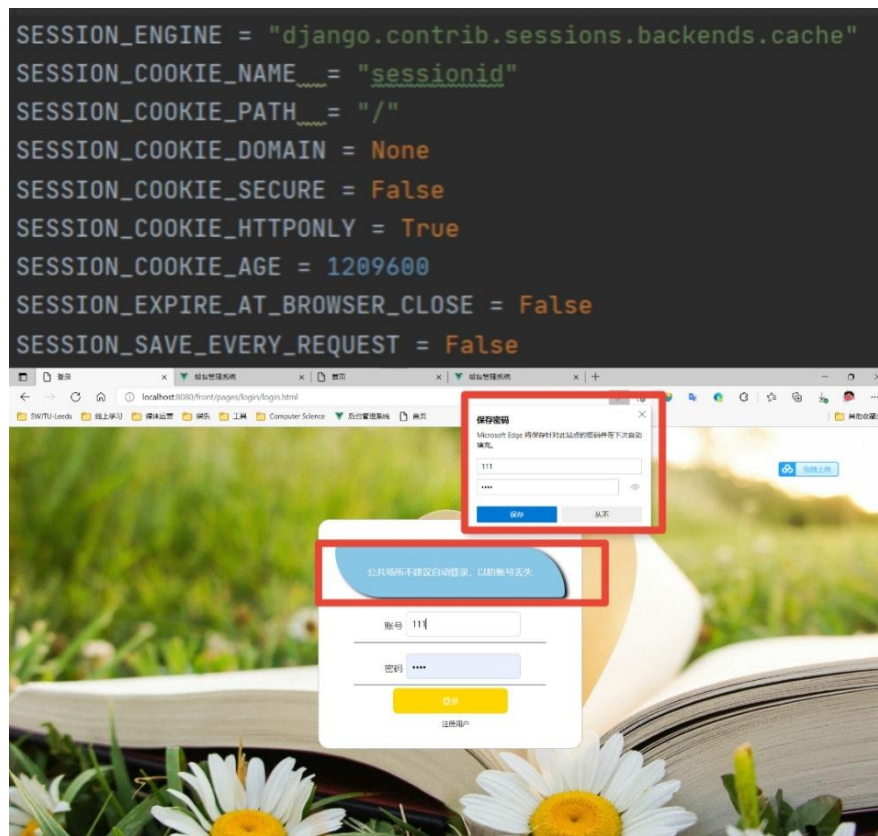
The main content area shows a table of published reading impressions. The table has columns for 索引 (Index), 标题 (Title), 图书名称 (Book Name), 图书分类 (Book Category), 封面图片 (Cover Image), 发布时间 (Release Time), 用户账号 (User Account), 用户姓名 (User Name), and 操作 (Actions). The '操作' column contains buttons for 详情 (Details), 修改 (Modify), and 删除 (Delete).

Three reading impressions are listed:

索引	标题	图书名称	图书分类	封面图片	发布时间	用户账号	用户姓名	操作
1	把自己作为方法读后感	把自己作为方法	社科纪实		2021-12-07	123	小不点	详情 修改 删除
2	世界杂货店读后感	世界杂货店	科幻奇幻		2021-12-07	123	小不点	详情 修改 删除
3	推理悬疑评价	推理悬疑	推理悬疑		2021-12-07	123	小不点	详情 修改 删除

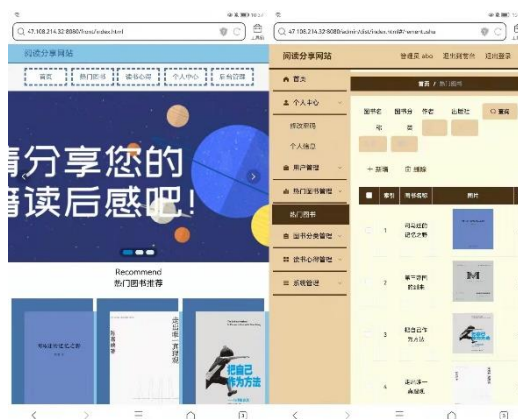
At the bottom, there is a pagination bar showing '共 3 条' (Total 3 items), '10 条/页' (10 items per page), and '前往 1 页' (Go to page 1).

- When the user logs in after registration, the browser will prompt the user whether to save the password for a privacy warning.



2.4 Appropriate styling

This webpage can also be viewed and used very well on a tablet.

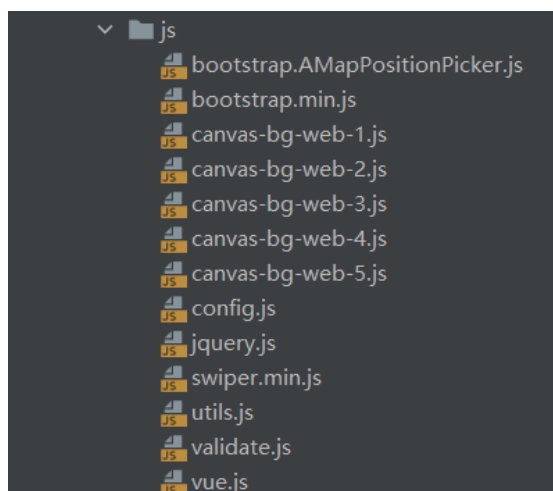


2.5 Deployment

- This website has been successfully deployed on the Internet.
- The login address of the background administrator is <http://47.108.214.32:8080/front/index.html>
The account is abo. The password is abo.
- The user access address of the front-end is <http://47.108.214.32:8080/front/index.html>

2.6 Features

- This website uses Bootstrap & jQuery in advanced features to enrich the front-end design of the web page.



3 Evaluation

3.1 Website Evaluation

- The evaluation of user experience relating ease of use back to design decisions.

Before designing this website, I browsed some of the same types of websites and summarized the similar web design structure, so I adopted a top-down layout. First, the main functional modules of the website are placed at the top of the page due to users tend to browse from top to bottom when browsing. Secondly, in addition to browsing the content on the website, users can also enjoy the basic functions provided by the website after registering an account, which also improves the user experience. For the color of the website, I mainly used blue, black, and yellow to avoid using too many colors to make users feel uncomfortable when browsing.



- The evaluation of potential security risk.
 - Server. Because the server itself is a system, the system may have problems with vulnerabilities, permissions, environments, and ports, and these problems will threaten the security of the web. In addition, new security issues will also arise because of the server's own configuration and authority service issues. Therefore, I used the Alibaba Cloud server, which provides a better server environment for my website, and provides me with some solutions when the server has problems.
 - Program and data. Vulnerabilities in the program itself can also pose a threat to the security of the web. For this reason, I adopt a testing method to reduce the vulnerabilities of web programs. In addition, when a user logs in in a public place, the web page will remind the user not to log in automatically, to reduce the threat of some harmful external network environment to this website.

3.2 Self-Evaluation

- In the process of building this website, I realized that compared with the same type of website, this website only provides some basic functions and lacks strong competitiveness.
- Based on the first coursework, in this project, I have a clearer understanding of website architecture. At the same time, it also allowed me to accumulate the experience of building a website and deploying it, which is very helpful to my study of this course.
- However, I still have a lot of abilities that need to be strengthened. For example, use more unit tests to test certain key functions of the website, and better use logs to record some useful information.

4 Reference

Kaplan-Moss, J., Holovaty, A. 2006. *The Django Book*. GNU Free Document License.

W3Cschool. *Django tutorial*. [Online]. [Accessed 20 November 2021]. Available from: <https://www.w3cschool.cn/django/>