



中山大學
SUN YAT-SEN UNIVERSITY

软件工程实验报告

实验课程: 软件工程

任课教师: 郑贵锋老师

实验题目: 开源软件分析与维护

专业名称: 计算机科学与技术

学生姓名: 林宏宇

学生学号: 23320093

实验时间: 2025 年 5 月 23 日

一、作业要求

此项实验报告包含了两篇综合实践的结果。

(1) 综合实践一

① 实践任务:选取或指定待阅读、分析和维护的开源软件。

② 实践方法:访问 GitHub、Gitee、SourceForge 等开源软件托管平台,从中检索符合上述要求的开源软件,下载或克隆开源软件代码,阅读开源软件的相关文档,安装、部署和运行开源软件。以二人为一组,采用结对方式开展本综合实践。

③ 实践要求:选取或指定的开源软件功能易于理解、代码质量高、规模适中(5 000~20 000 行代码量),也可以直接指定“小米便签”软件 MiNotes 作为阅读、分析和维护的对象。

④ 实践结果:获得开源软件的源代码,并可运行和操作该开源软件。

1. 综合实践一

实践任务:理解和分析开源软件的整体情况。

实践方法:运行和使用开源软件,理解软件功能;泛读开源代码,分析代码的构成,包括:有哪些子系统、模块,模块与功能的对应关系,软件模块间的关系等,在此基础上绘制软件系统的体系结构图(可以用 UML 的包图和类图来描述);利用 SonarQube 工具分析开源代码的质量情况。

实践要求:理解开源软件提供的功能和服务,掌握软件系统的模块构成,分析开源软件的质量水平。

实践结果:① 软件需求文档,描述开源软件的大致需求;② 软件体系结构图,描述开源软件的模块构成;

③ SonarQube 的开源软件质量报告。

二、实验背景介绍

本实验报告是软件工程学科综合实践任务的项目作业,目的是通过分析和维护开源软件,掌握开源项目的基本结构和运行机制,加强对学科的理解,提升对开源软件的理解和实际操作能力。

通过阅读和分析开源代码,能够深入理解并学习到一些优秀的编程实践和设计模式,从而提升个人编程能力。在实际操作过程中,能够掌握项目部署、调试和优化的技巧,进一步提高自己的技术水平。同时,通过维护和改进开源项目,不仅能够为开源社区贡献力量,还能推动技术的发展和进步。此外,参与开源项目的过程中,能够锻炼团队协作能力,学会使用版本控制工具,从而提高自己的协作能力和团队合作经验。

三、实践任务

在本次综合实践任务中,可通过该项目学习 Django 框架、DRF 接口开发以及微信小程序的集成开发。

1. 选择的开源软件名称:SH 校园二手商城项目(基于 BaykeShop 拜客商城)
2. 开源项目链接: <https://github.com/3BPM/shshop.git>
3. 开源软件托管平台:Github
4. 目标用户群体:校园内的学生,主要用于二手商品的发布和交易。
5. 开源软件概述:SH 校园二手商城项目是一个基于 BaykeShop 的二次开发项目,旨在为校园用户提供一个二手商品交易平台。其主要功能包括:用户发布商品和求购信息、商品分类管理(基于校区标签)、用户余额

管理和信息管理、微信小程序端与后端的统一交互。

四、实践方法

(一) 克隆开源软件的源代码:

从 github 上 clone 项目到本地, 使用 vscode 打开进行项目阅读。

(二) 软件的功能需求:

这个项目是基于 BaykeShop 拜客商城改进的, 增加了二手交易功能, 并支持微信小程序。使用 Django+DRF 作为后端, Vue 作为前端, 并且涉及 MySQL、Redis、Nginx、uWSGI 等服务。

(三) 软件实现:

1. 用户可以发布商品和求购信息。
2. 微信小程序端与后端实现多端统一, 用户可通过小程序查看商品和求购信息。
3. 商品分类基于校区标签, 方便校园用户快速筛选。
4. 提供用户余额管理和信息管理功能。
5. 后端基于 Django REST Framework (DRF), 但安全性较为简陋, 需进一步优化。

(四) 运行环境与依赖:

1. Python 版本: 大于等于 3.8
2. 框架: Django 4.1, 并使用 Django Rest Framework 来构建 API。
3. 数据库: MySQL 8.0
4. 缓存: Redis
5. 服务器部署: Nginx + uWSGI
6. 前端使用 Vue

(五) Django 项目:

Django 是一个高级的 Python Web 框架, 旨在以最小的代码量实现快速开发、安全性高、可扩展性强的 Web 应用。本次实验的 Django 项目基于 Model-View-Template 架构, 提供了完整的数据库交互、后台管理、用户认证等功能, 适用于快速构建 Web 应用。

项目特点:

1. 基于 MVT 架构
 - 1) Model 模型层: 负责与数据库交互, 定义数据结构, 并提供 ORM 对象关系映射支持, 使开发者可以使用 Python 代码操作数据库, 而无需编写 SQL 语句。
 - 2) View 视图层: 处理业务逻辑, 接收请求、处理数据并返回响应。
 - 3) Template 模板层: 用于前端页面的渲染, 结合 Django 的模板语言, 实现数据的动态展示。
2. 内置功能

- 1) 用户认证与权限管理: Django 自带完整的用户管理系统, 包括登录、注册、权限控制等功能。
 - 2) 后台管理系统: Django Admin 提供了强大的后台管理界面, 便于数据管理。
 - 3) 路由与 URL 解析: 支持基于正则表达式或 path 方式的 URL 设计, 使路由管理更加灵活。
 - 4) 表单处理: Django 提供了强大的表单组件, 可自动处理表单验证, 提高开发效率。
3. 高效开发
- 1) Django 采用 DRY 原则, 减少代码重复, 提高可维护性。
 - 2) 内置 SQLite、MySQL、PostgreSQL 等数据库支持, 便于数据存储与管理。
 - 3) 采用 MTV 设计模式, 使代码结构清晰, 便于扩展和维护。

五、环境配置安装、部署和运行软件

(一) 创建并激活虚拟环境

创建虚拟环境

```
cd sh-shop  
python3 -m venv venv
```

Windows 激活虚拟环境

```
Windows: venv\Scripts\activate
```

激活结果显示

```
PS D:\Project_Shop\shshop> Get-ExecutionPolicy  
Restricted  
PS D:\Project_Shop\shshop> Set-ExecutionPolicy RemoteSigned  
PS D:\Project_Shop\shshop> venv\Scripts\activate  
(venv) PS D:\Project_Shop\shshop>
```

安装依赖

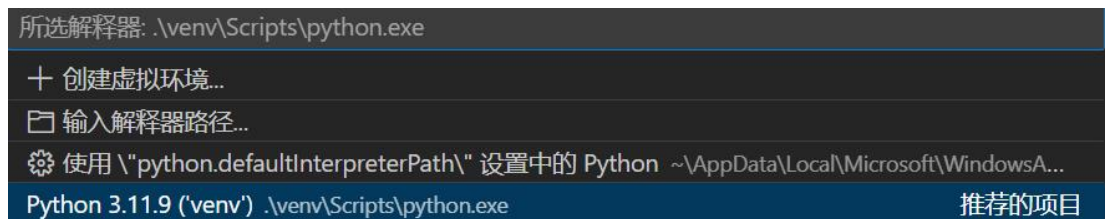
```
pip install -r requirements.txt
```

安装结果显示

```
[notice] A new release of pip is available: 24.0 -> 25.0.1  
[notice] To update, run: python.exe -m pip install --upgrade pip  
(venv) PS D:\Project_Shop\shshop>
```

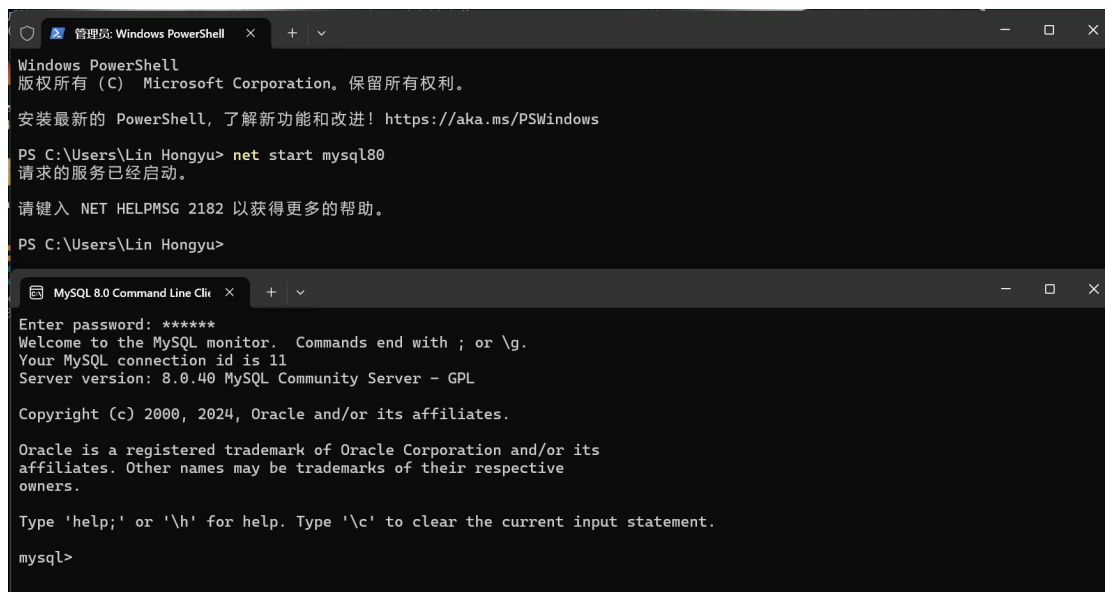
(二) Django 框架 python 解释器

在 vscode 选择 venv 的 Python 解释器(不然 django 代码报错)



(三) 安装 MySQL 和 Redis

安装结果显示



(四) 配置 MySQL 和 Redis

注意设置账户、密码、端口、ip 等信息



```
CACHES = {
    "default": {
        "BACKEND": "django.core.cache.backends.redis.RedisCache",
        "LOCATION": "redis://127.0.0.1:6379",
    }
}
```

(五) 同步数据库

```
(venv) PS D:\Project_Shop\shshop> python manage.py makemigrations
No changes detected
(venv) PS D:\Project_Shop\shshop> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, authtoken, contenttypes, flatpages, sessions, shshop, sites, videos
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying authtoken.0001_initial... OK
  Applying authtoken.0002_auto_20160226_1747... OK
  Applying authtoken.0003_tokenproxy... OK
  Applying sites.0001_initial... OK
  Applying flatpages.0001_initial... OK
  Applying sessions.0001_initial... OK
  Applying shshop.0001_initial... OK
  Applying shshop.0002_initial... OK
  Applying sites.0002_alter_domain_unique... OK
  Applying videos.0001_initial... OK
(venv) PS D:\Project_Shop\shshop>
```

(六) 创建超级管理员，使用时需要创建账户 (端口加上 127.0.0.1:不然出错)

```
PS D:\> cd Project_Shop\shshop
PS D:\Project_Shop\shshop> venv\Scripts\activate
(venv) PS D:\Project_Shop\shshop> python manage.py createsuperuser
用户名 (leave blank to use 'linhongyu'): python manage.py push
Error: 请输入合法的用户名。只能包含英文字母、数字、特殊字符“@”、“.”、“_”和“-”。
用户名 (leave blank to use 'linhongyu'): full
电子邮件地址: 1289421662@qq.com
Password:
Password (again):
Superuser created successfully.
(venv) PS D:\Project_Shop\shshop> python manage.py push
Successfully "push shadmin menus"
(venv) PS D:\Project_Shop\shshop> python manage.py push -test
Successfully "push shadmin menus"
(venv) PS D:\Project_Shop\shshop> python manage.py runserver 0:8000
Performing system checks...

System check identified no issues (0 silenced).
Error: [Errno 11001] getaddrinfo failed
(venv) PS D:\Project_Shop\shshop> python manage.py runserver 127.0.0.1:8000
Performing system checks...

System check identified no issues (0 silenced).
March 18, 2025 - 08:19:08
Django version 4.2.4, using settings 'sh.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

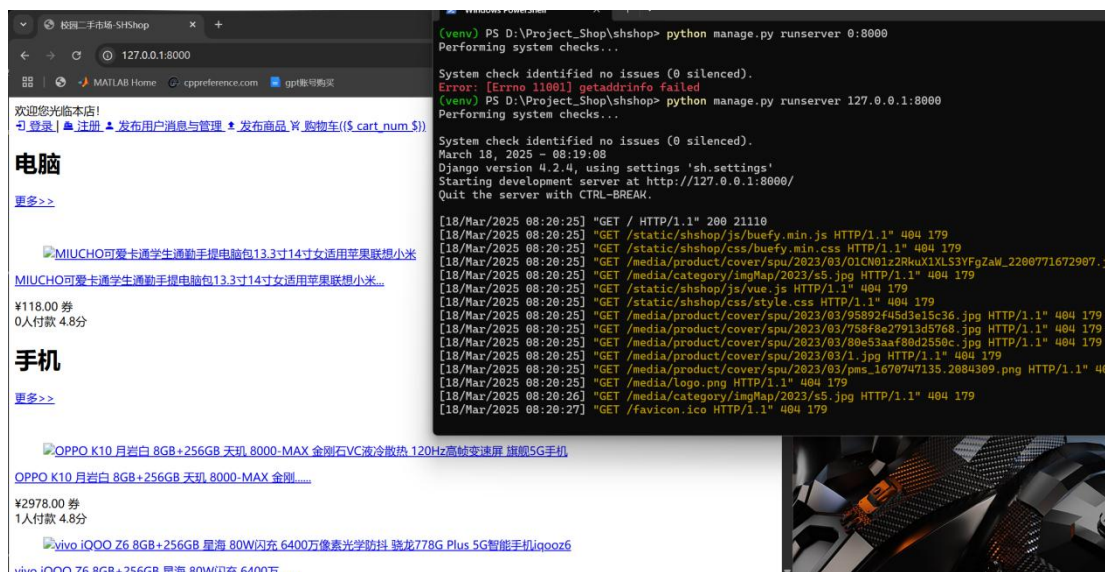
(七) 启动项目


```
PS D:\> cd Project_Shop\shshop
PS D:\Project_Shop\shshop> venv\Scripts\activate
(venv) PS D:\Project_Shop\shshop> python manage.py createsuperuser
用户名 (Leave blank to use 'linhongyu'): python manage.py push
Error: 请输入合法的用户名。只能包含英文字母、数字、特殊字符“@”、“.”、“-”和“_”。
用户名 (Leave blank to use 'linhongyu'): full
电子邮件地址: 1289421662@qq.com
Password:
Password (again):
Superuser created successfully.
(venv) PS D:\Project_Shop\shshop> python manage.py push
Successfully "push shadmin menus"
(venv) PS D:\Project_Shop\shshop> python manage.py push -test
Successfully "push shadmin menus"
(venv) PS D:\Project_Shop\shshop> python manage.py runserver 0:8000
Performing system checks...

System check identified no issues (0 silenced).
Error: [Errno 11001] getaddrinfo failed
(venv) PS D:\Project_Shop\shshop> python manage.py runserver 127.0.0.1:8000
Performing system checks...

System check identified no issues (0 silenced).
March 18, 2025 - 08:19:08
Django version 4.2.4, using settings 'sh.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

可以通过网站访问项目，接过显示(前端图片加载不出来)



六、部署配置

- (一) 下载并配置 uwsgi
 - (Windows 不太支持 uwsgi，这一步操作较麻烦)
 - 在激活虚拟环境情况下下载 uwsgi，报错！

```
管理员: Windows PowerShell
(venv) PS D:\Project_Shop\shshop> pip install uwsgi
Collecting uwsgi
  Downloading uwsgi-2.0.28.tar.gz (816 kB)
    816.2/816.2 kB 2.2 MB/s eta 0:00:00
Installing build dependencies ... done
Getting requirements to build wheel ... error
error: subprocess-exited-with-error

× Getting requirements to build wheel did not run successfully.
  exit code: 1
  > [23 lines of output]
```

Uwsgi.ini 文件配置

```
uwsgi.ini
1  [uwsgi]
2  socket = 127.0.0.1:9090
3  wsgi-file=/root/shshop/sh/wsgi.py
4  master = true
5  processes = 4
6  chmod-socket = 666
7  vacuum = true
8  chdir = /root/shshop
9  py-autoreload = 1
10
```

Windows 支持不好，使用 Cygwin 终端

```
/cygdrive/d/uwsgi-2.0.28
Lin Hongyu@LAPTOP-RDA9S5SE ~
$ cd /cygdrive/d/uwsgi-2.0.28
Lin Hongyu@LAPTOP-RDA9S5SE /cygdrive/d/uwsgi-2.0.28
$ python setup.py install
/usr/lib/python3.9/site-packages/setuptools/_distutils/dist
```

uwsgi 下载成功，但总显示缺失 Django 模板，两者连接不上

```
/cygdrive/d/uwsgi-2.0.28
Lin Hongyu@LAPTOP-RDA9S5SE /cygdrive/d/uwsgi-2.0.28
$ uwsgi -version
*** Starting uWSGI 2.0.28 (64bit) on [Tue Mar 18 10:56:04 2025] ***
compiled with version: 12.4.0 on 18 March 2025 02:29:02
os: CYGWIN_NT-10.0-26100-3.5.7-1.x86_64 2025-01-29 19:46 UTC
nodename: LAPTOP-RDA9S5SE
machine: x86_64
clock source: unix
pcre jit disabled
detected number of CPU cores: 16
current working directory: /cygdrive/d/uwsgi-2.0.28
detected binary path: /usr/bin/uwsgi
*** WARNING: you are running uWSGI without its master process manager ***
your memory page size is 4096 bytes
detected max file descriptor number: 3200
lock engine: windows mutexes
thunder lock: disabled (you can enable it with --thunder-lock)
The -s/--socket option is missing and stdin is not a socket.
```

再次在 Cygwin 运行指令，还是失败..!!!

```
uwsgi --ini uwsgi.ini
```


需要转换命令的格式，在 Cygwin 启动虚拟环境

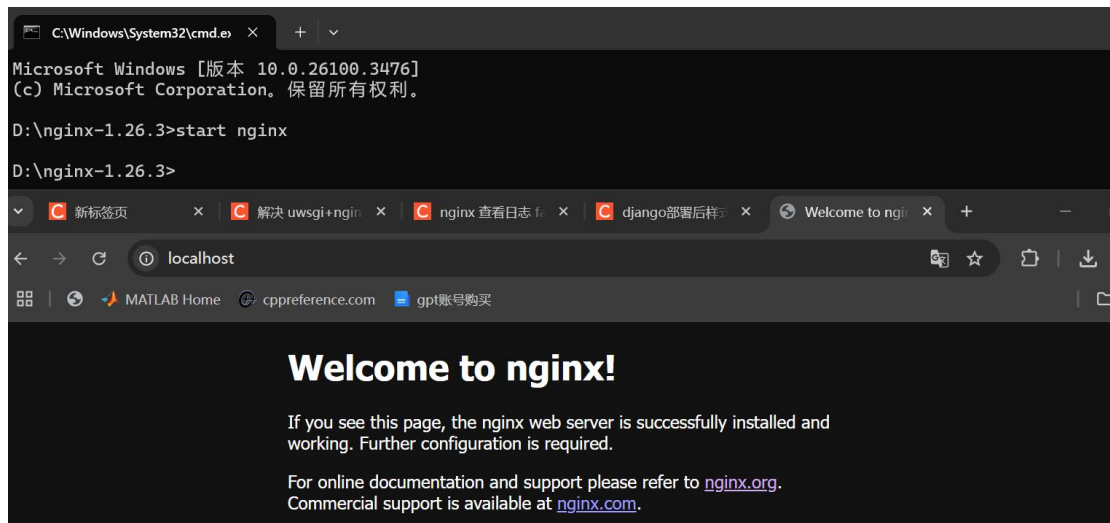
```
/cygdrive/d/Project_Shop/shshop
Lin Hongyu@LAPTOP-RDA9S5SE ~
$ cd D:
Lin Hongyu@LAPTOP-RDA9S5SE /cygdrive/d
$ cd Project_Shop
Lin Hongyu@LAPTOP-RDA9S5SE /cygdrive/d/Project_Shop
$ cd shshop
Lin Hongyu@LAPTOP-RDA9S5SE /cygdrive/d/Project_Shop/shshop
$ dos2unix venv/Scripts/activate
dos2unix: converting file venv/Scripts/activate to Unix format...
Lin Hongyu@LAPTOP-RDA9S5SE /cygdrive/d/Project_Shop/shshop
$ source venv/Scripts/activate
(venv)
Lin Hongyu@LAPTOP-RDA9S5SE /cygdrive/d/Project_Shop/shshop
$ python manage.py collectstatic
You have requested to collect static files at the destination
location as specified in your settings:
    D:\Project_Shop\shshop\static
This will overwrite existing files!
Are you sure you want to do this?
Type 'yes' to continue, or 'no' to cancel: yes
0 static files copied to 'D:\Project_Shop\shshop\static', 2533 unmodified.
(venv)
Lin Hongyu@LAPTOP-RDA9S5SE /cygdrive/d/Project_Shop/shshop
$ |
```

运行还是报错，这一次可以启动，但访问网站依旧报错

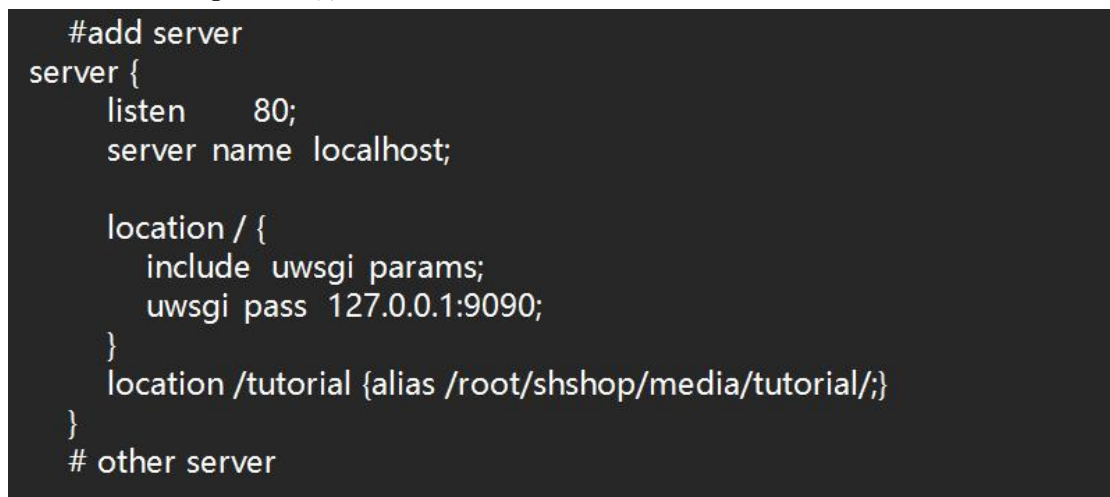
```
/cygdrive/d/Project_Shop/shshop
Lin Hongyu@LAPTOP-RDA9S5SE /cygdrive/d/Project_Shop
$ cd shshop
Lin Hongyu@LAPTOP-RDA9S5SE /cygdrive/d/Project_Shop/shshop
$ uwsgi --ini uwsgi.ini
[uWSGI] getting INI configuration from uwsgi.ini
*** Starting uWSGI 2.0.28 (64bit) on [Wed Mar 19 17:13:18 2025] ***
compiled with version: 12.4.0 on 18 March 2025 02:29:02
os: CYGWIN_NT-10.0-26100-3.5.7-1.x86_64 2025-01-29 19:46 UTC
nodename: LAPTOP-RDA9S5SE
machine: x86_64
clock source: unix
pcre jit disabled
detected number of CPU cores: 16
current working directory: /cygdrive/d/Project_Shop/shshop
detected binary path: /usr/bin/uwsgi
chdir() to D:/Project_Shop/shshop
your memory page size is 4096 bytes
detected max file descriptor number: 3200
lock engine: windows mutexes
thunder lock: disabled (you can enable it with --thunder-lock)
uwsgi socket 0 bound to TCP address 127.0.0.1:9090 fd 3
Python version: 3.9.16 (main, Mar 8 2023, 22:47:22) [GCC 11.3.0]
Python main interpreter initialized at 0xa0002eb70
python threads support enabled
your server socket listen backlog is limited to 100 connections
your mercy for graceful operations on workers is 60 seconds
mapped 364520 bytes (355 KB) for 4 cores
*** Operational MODE: preforking ***
Traceback (most recent call last):
  File "D:/Project_Shop/shshop/sh/wsgi.py", line 12, in <module>
    from django.core.wsgi import get_wsgi_application
ModuleNotFoundError: No module named 'django'
unable to load app 0 (mountpoint='') (callable not found or import error)
*** no app loaded. going in full dynamic mode ***
*** uWSGI is running in multiple interpreter mode ***
spawned uWSGI master process (pid: 1342)
spawned uWSGI worker 1 (pid: 1343, cores: 1)
spawned uWSGI worker 2 (pid: 1344, cores: 1)
Python auto-reloader enabled
spawned uWSGI worker 3 (pid: 1345, cores: 1)
spawned uWSGI worker 4 (pid: 1346, cores: 1)
```

(二) Nginx 下载并配置

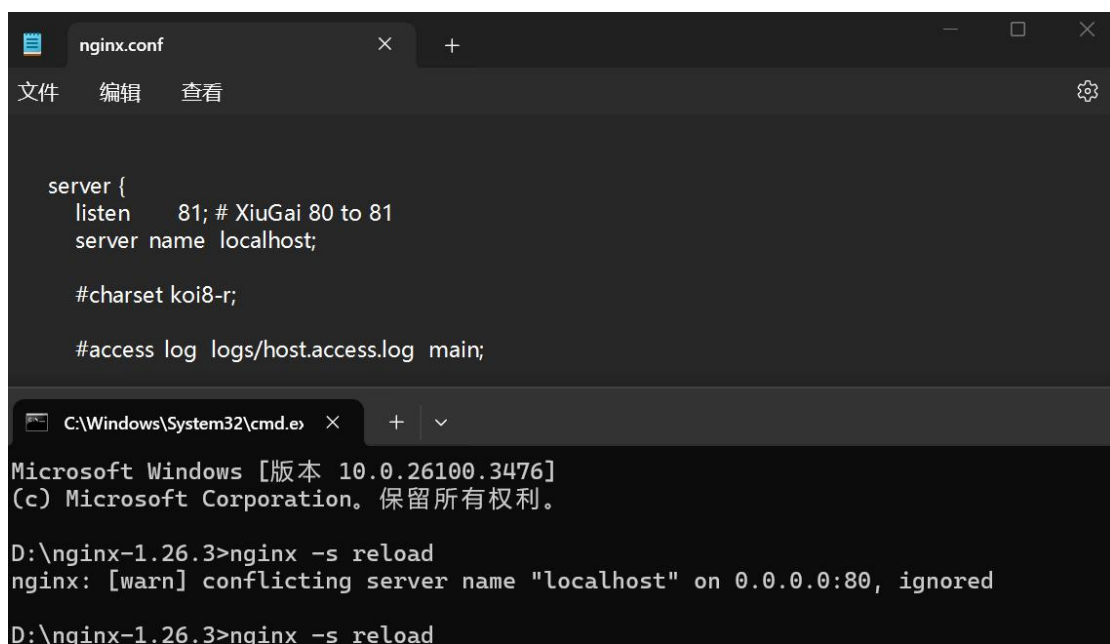
可以通过网页访问，成功



配置 Nginx 文件



端口与默认端口冲突 修改原默认配置 重新启动



（三）Django 会将所有静态文件收集到指定的目录中

```
管理员: Windows PowerShell
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！ https://aka.ms/PSWindows

PS C:\Users\Lin Hongyu> D:
PS D:\> cd Project_Shop\shshop
PS D:\Project_Shop\shshop> venv\Scripts\activate
(venv) PS D:\Project_Shop\shshop> python manage.py collectstatic

You have requested to collect static files at the destination
location as specified in your settings:

    D:\Project_Shop\shshop\static

This will overwrite existing files!
Are you sure you want to do this?

Type 'yes' to continue, or 'no' to cancel: yes

2446 static files copied to 'D:\Project_Shop\shshop\static', 87 unmodified.
```

七、开源软件的整体分析

项目的整体结构

Django 项目目录结构：

shshop 项目根目录

(1) manage.py

Django 项目的入口文件，用于运行服务器、执行数据库迁移等管理任务。

(2) requirements.txt

列出了项目所需的 Python 包及其版本，用于安装依赖。

(3) uwsgi.ini

uWSGI 配置文件，用于部署

(4) README.md

项目说明文档

(5) Static

静态文件目录（由 collectstatic 收集）

(6) Media

媒体文件目录（如用户上传的文件）

(7) Sh

主应用目录（项目配置）

1) __init__.py

Python 包标识文件

2) settings.py

项目的全局配置文件，包括数据库、缓存、静态文件等配置。

3) urls.py

- 全局路由配置
- 4) wsgi.py
WSGI 入口文件，用于部署
- 5) asgi.py
ASGI 入口文件（如果使用异步支持）
- (8) Shshop
具体的 Django 应用目录
 - 1) Config
可能存放项目的配置文件或工具类
 - 2) Management
Django 的管理命令目录，用于定义自定义管理命令
 - 3) Public
公共模块，可能包含通用的模型、工具或功能
 - 4) Static
应用的静态文件目录
 - 5) __init__.py
Python 包标识文件
 - 6) admin.py
管理后台配置
 - 7) apps.py
应用配置
 - 8) Migrations
数据库迁移文件
 - 9) Module
数据库模型定义
 - 10) tests.py
单元测试
 - 11) views.py
视图逻辑
 - 12) urls.py
应用的路由配置
 - 13) Templates
HTML 模板文件
- (9) Api
API 接口模块，使用 Django REST Framework 实现，提供后端接口。
- (10) sh-fronted 微信小程序
微信小程序前端代码目录，用于实现用户界面和交互。
- (11) Utils
工具模块，可能包含通用的工具函数或类
- (12) Venv
Python 虚拟环境目录，存放项目的依赖包
- (13) Videos
与视频相关的功能模块
- (14) nginxconfbackup

八、主要模块分析

（一）配置文件 `setting.py`

1. 基本信息：定义根目录路径
2. 安全配置
3. 已安装的应用：第三方管理后台、用户认证、内容类型框架等以及自订的引用 `shshop` 和 `videos` 等
4. 中间件：处理请求和响应的钩子按顺序执行
5. 路由配置：指定文件 `urls.py`
6. 模板配置
7. WSGI 配置：指定 WSGI 入口文件，用于部署
8. 数据库配置：MySQL
9. 国际化配置：语言、时区等
10. 静态文件和媒体文件：指定配置的 URL 和存储路径
11. 缓存配置：Redis
12. 邮件配置：指定 QQ 邮箱密码和授权码
13. REST Framework 配置：配置 Django REST Framework 的默认模式类，用于生成 API 文档
14. SimpleUI 配置：主题、首页、logo 等

（二）URL 路由规则 `urls.py`

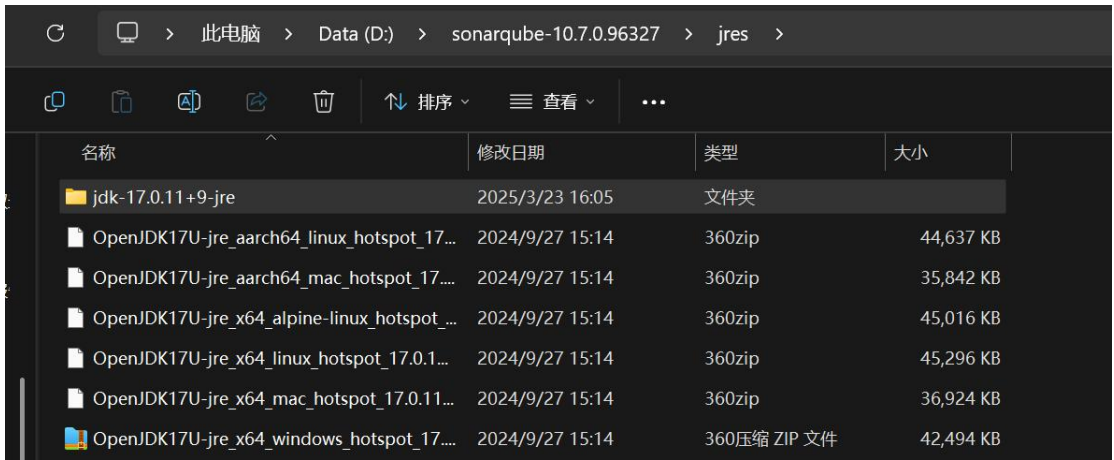
1. 映射 Django 自带的管理后台路径
2. 映射自带的管理后台站点
3. 映射应用 `shshop` 的路由配置
4. 映射管理静态页面的路径
5. 映射 `api` 的路由配置
6. 映射 Django REST Framework 的 API
7. 设置在开发者模式下提供静态服务
8. 设置开发模式下为媒体文件提供服务

（三）视图逻辑 `views.py`

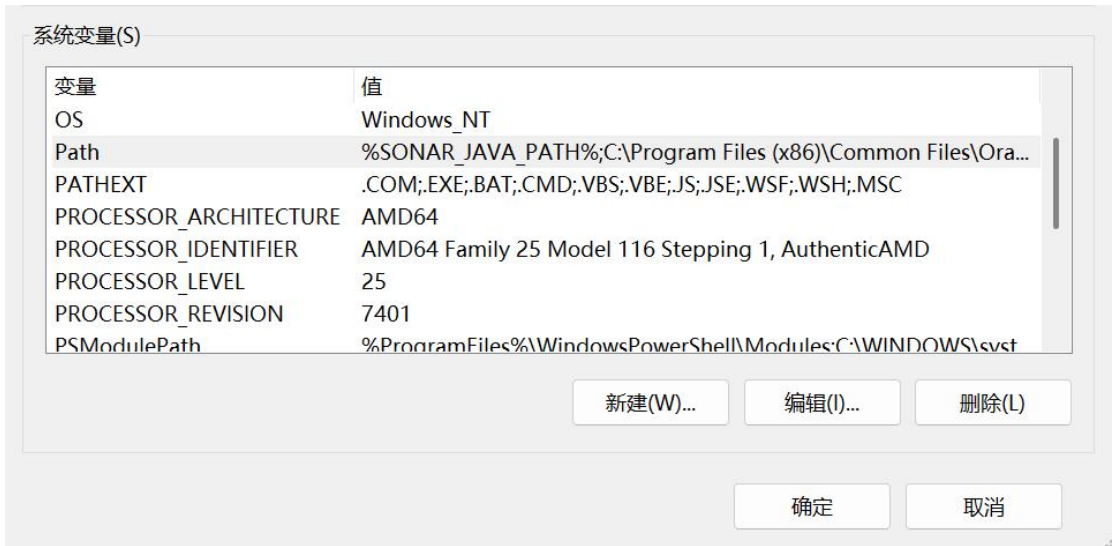
1. 概览：使用 Django REST Framework 的一个通用试图类，用于处理列表查询和创建操作
2. 通用视图结构：视图操作的模型数据集和序列化器
3. 功能包括：菜单、权限、轮播图、用户信息、尚品、订单、文章等核心模块的 API 接口

九、SonarQube+SonarScanner 下载配置

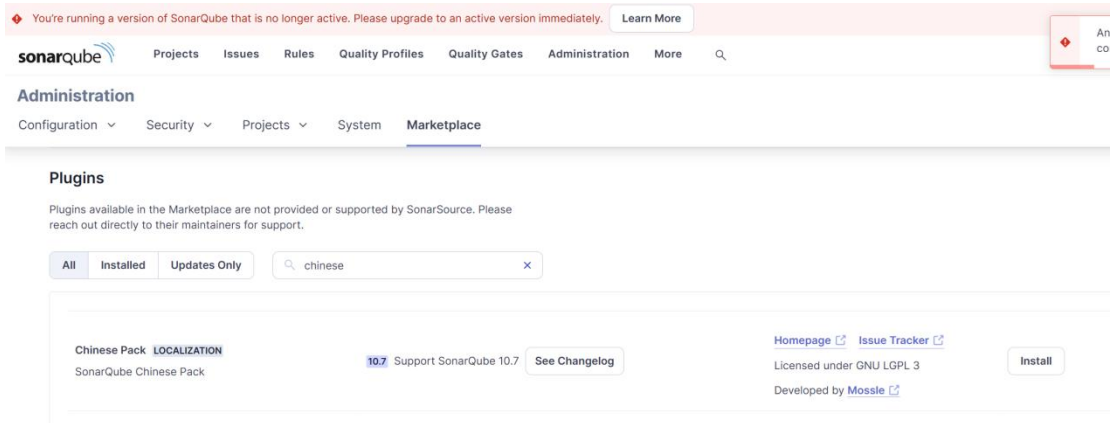
（一）下载 SonarQube



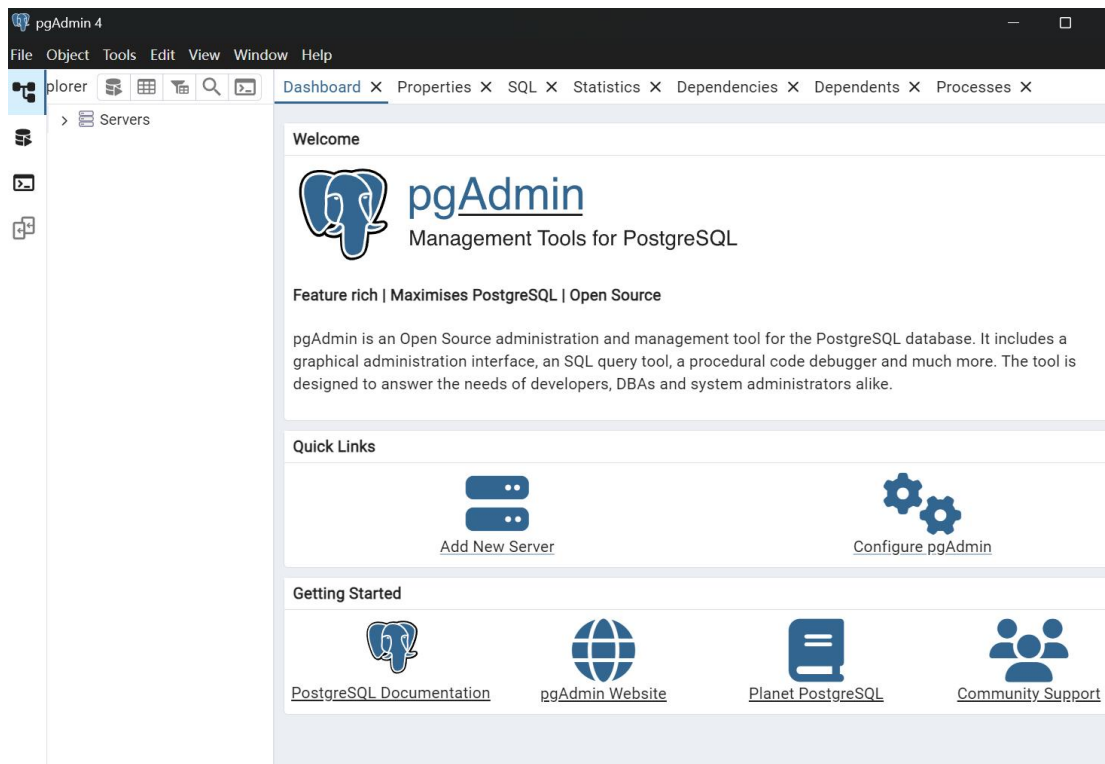
（二）搭配环境变量



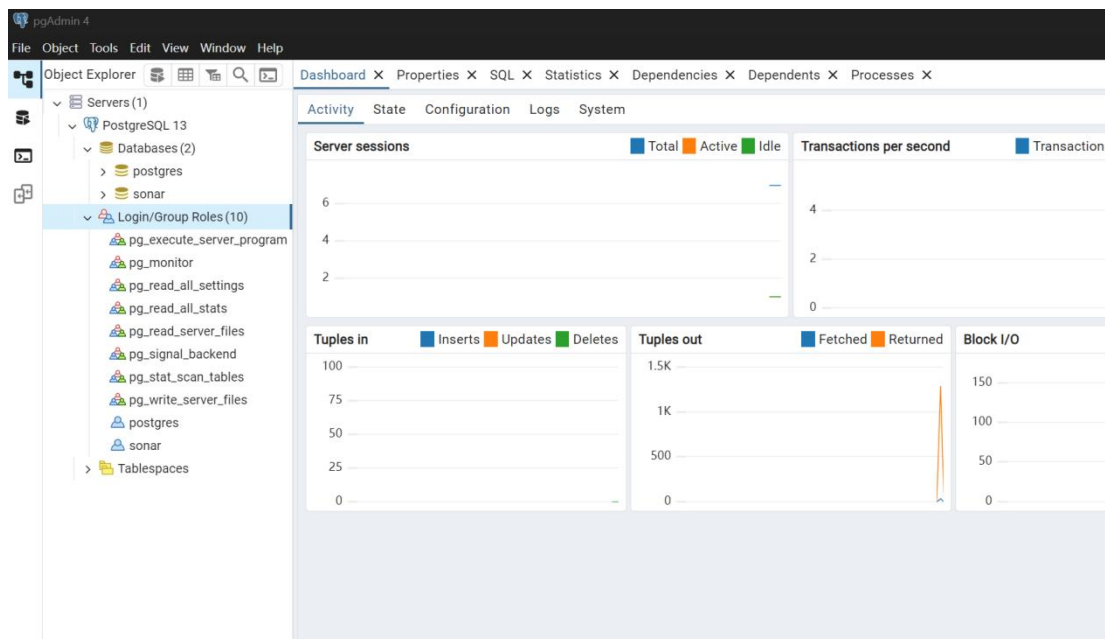
（三）SonarQube 可以运行，下载汉化插件



（四）搭配数据库 postgresQL



(五) postgresQL 配置账户和环境



(六) 连接 postgresQL 和 SonarQube，配置环境

```

22 # User credentials.
23 # Permissions to create tables, indices and triggers must be granted to JDBC user.
24 # The schema must be created first.
25 sonar.jdbc.username=sonar
26 sonar.jdbc.password=sonar
27
28 #----- Embedded Database (default)
29 # H2 embedded database server listening port, defaults to 9092
30 #sonar.embeddedDatabase.port=9092
31
32
33 #----- Oracle 19c/21c
34 # The Oracle JDBC driver must be copied into the directory extensions/jdbc-driver/oracle/.
35 # Only the thin client is supported, and we recommend using the latest Oracle JDBC driver.
36 # https://jira.sonarsource.com/browse/SONAR-9758 for more details.
37 # If you need to set the schema, please refer to http://jira.sonarsource.com/browse/SONAR-5
38 #sonar.jdbc.url=jdbc:oracle:thin:@localhost:1521/XE
39
40
41 #----- PostgreSQL 11 or greater
42 # By default the schema named "public" is used. It can be overridden with the parameter "currentSchema"
43 sonar.jdbc.url=jdbc:postgresql://localhost/sonar?currentSchema=public

```

(七) 创建代码分析项目

配置 sonar-project.properties 文件

```

sonar-project.properties X
D: > Shop_SonarTest > sonar-project.properties
1 # must be unique in a given SonarQube instance
2 sonar.projectKey=sh-test
3
4 # --- optional properties ---
5
6 # defaults to project key
7 sonar.projectName=sh-test
8 # defaults to 'not provided'
9 #sonar.projectVersion=1.0
10
11 # Path is relative to the sonar-project.properties file. Defaults to .
12 sonar.sources=.
13
14 # Encoding of the source code. Default is default system encoding
15 sonar.sourceEncoding=UTF-8
16
17 sonar.token=squ_6ca3410a8991be829706ddbc8c34ff565319a518

```

(八) 运行并观察结果

```

PS D:\Shop_SonarTest> sonar-scanner
20:13:05.794 INFO Scanner configuration file: D:\sonar-scanner-7.0.2.4839-windows-x64\bin\..\conf\sonar-scanner.properties
20:13:05.800 INFO Project root configuration file: D:\Shop_SonarTest\sonar-project.properties
20:13:05.821 INFO SonarScanner CLI 7.0.2.4839
20:13:05.823 INFO Java 17.0.13 Eclipse Adoptium (64-bit)
20:13:05.823 INFO Windows 11 10.0 amd64
20:13:05.840 INFO User cache: C:\Users\Lin Hongyu\.sonar\cache
20:13:06.481 INFO JRE provisioning: os[windows], arch[amd64]
20:13:08.739 INFO Communicating with SonarQube Server 10.7.0.96327
20:13:09.234 INFO Starting SonarScanner Engine...
20:13:09.235 INFO Java 17.0.11 Eclipse Adoptium (64-bit)
20:13:10.054 INFO Load global settings
20:13:10.124 INFO Load global settings (done) | time=69ms
20:13:10.128 INFO Server id: A7EE8CF2-AZXCPTnSbt6jSV0Bvb0
20:13:10.135 INFO Loading required plugins
20:13:10.136 INFO Load plugins index
20:13:10.161 INFO Load plugins index (done) | time=25ms
20:13:10.161 INFO Load/download plugins
20:13:10.754 INFO Load/download plugins (done) | time=593ms
20:13:11.121 INFO Process project properties
20:13:11.147 INFO Process project properties (done) | time=25ms
20:13:11.161 INFO Project key: sh-test
20:13:11.162 INFO Base dir: D:\Shop_SonarTest
20:13:11.162 INFO Working dir: D:\Shop_SonarTest\.scannerwork

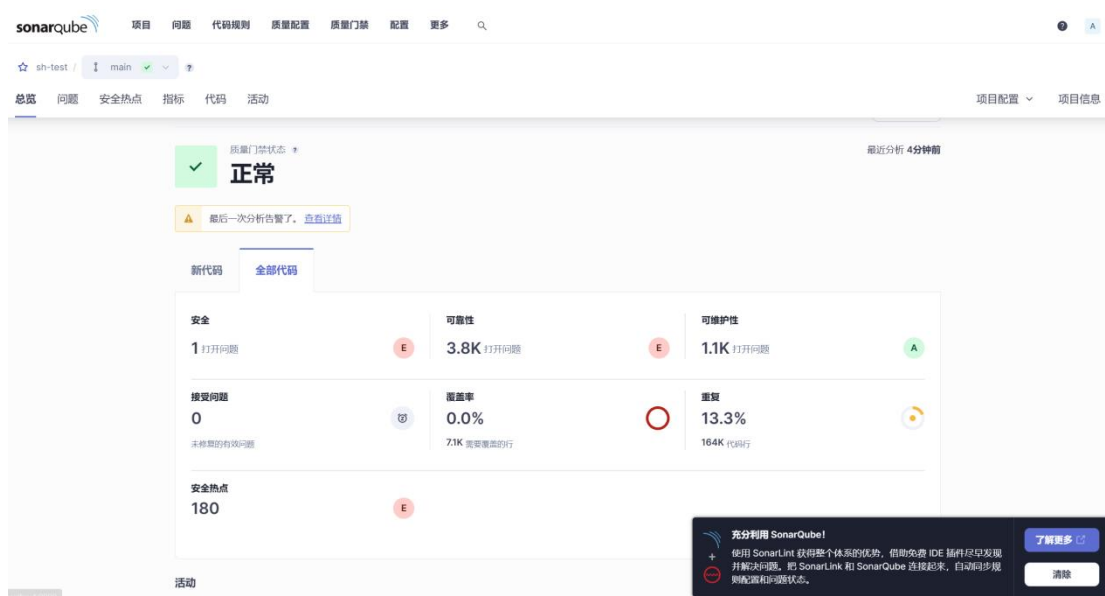
```

运行完成

```
Windows PowerShell
20:15:16.824 WARN Too many duplication groups on file NoGit_shop/media/tutorial/C04-View/Routers.html. Keep only the first 100 groups.
20:15:16.837 WARN Too many duplication groups on file NoGit_shop/media/tutorial/C04-View/RequestAndResponse.html. Keep only the first 100 groups.
20:15:16.856 WARN Too many duplication groups on file NoGit_shop/media/tutorial/C02-DRFProject/index.html. Keep only the first 100 groups.
20:15:16.869 WARN Too many duplication groups on file NoGit_shop/media/tutorial/C04-View/ViewSet.html. Keep only the first 100 groups.
20:15:16.897 WARN Too many duplication groups on file NoGit_shop/media/tutorial/C05-Components/index.html. Keep only the first 100 groups.
20:15:16.916 WARN Too many duplication groups on file NoGit_shop/media/tutorial/C01-IntroduceToDRF/index.html. Keep only the first 100 groups.
20:15:16.934 WARN Too many duplication groups on file NoGit_shop/media/tutorial/C05-Components/Pagination.html. Keep only the first 100 groups.
20:15:16.950 WARN Too many duplication groups on file NoGit_shop/media/tutorial/C03-Serializer/Serializing.html. Keep only the first 100 groups.
20:15:16.952 INFO CPD Executor CPD calculation finished (done) | time=1097ms
20:15:24.885 INFO Analysis report generated in 505ms, dir size=9.4 MB
20:15:28.226 INFO Analysis report compressed in 3331ms, zip size=3.2 MB
20:15:29.139 INFO Analysis report uploaded in 913ms
20:15:29.142 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sh-test
20:15:29.142 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
20:15:29.143 INFO More about the report processing at http://localhost:9000/api/ce/task?id=4b2df894-65bc-4aa9-a709-06f8db105523
20:15:37.840 INFO Analysis total time: 2:26.999 s
20:15:37.841 INFO SonarScanner Engine completed successfully
20:15:38.341 INFO EXECUTION SUCCESS
20:15:38.342 INFO Total time: 2:32.550s
PS D:\Shop_SonarTest>
```

十、SonarQube 代码质量分析

（一）查看 SonarQube 分析结果



Make sure this Django key gets revoked, changed, and removed from the code. [🔗](#)

Django secret keys should not be disclosed [secrets:S6687](#)

影响行 L23 · 影响: 30min · 介绍 10分钟前 · [🔒 漏洞](#) · [🛑 阻断](#)

干净代码属性

责任性 | 不值得信赖

软件质量影响:

安全 [🔴](#)

打开 ▾ 未分配 ▾ cwe [+](#)

问题的位置 问题原因 修复方法 活动 更多信息

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

Quick-start development settings - unsuitable for production

See <https://docs.djangoproject.com/en/4.1/howto/deployment/checklist/>

SECURITY WARNING: keep the secret key used in production secret!

SECRET_KEY = "django-insecure-#(\$0@z-)#8skc_qn-%p17b_isl8t+64q2z6bd5c)h^h=)l2%+"

Make sure this Django key gets revoked, changed, and removed from the code.

SECURITY WARNING: don't run with debug turned on in production!

DEBUG = False

ALLOWED_HOSTS = ["*"]

Application definition

INSTALLED_APPS = [

NoGit_shop/shshop/static/shshop/css/materialdesignicons.css

☐ Unexpected missing generic font family

目的性

可靠性 [🔴](#)

无标签 [+](#)

打开 ▾ 未分配 ▾

L13 · 1min 影响 · 11分钟前 · [🐞 Bug](#) · [🔴 主要](#)

NoGit_shop/shshop/templates/dev.html

☐ Add "lang" and/or "xml:lang" attributes to this "<html>" element

目的性

可靠性 [🔴](#)

accessibility wcag2-a [+](#)

打开 ▾ 未分配 ▾

L2 · 2min 影响 · 11分钟前 · [🐞 Bug](#) · [🔴 主要](#)

NoGit_shop/shshop/templates/shshop/cart/carts.html

☐ Add an "alt" attribute to this image.

目的性

可靠性 [🟡](#)

accessibility wcag2-a [+](#)

打开 ▾ 未分配 ▾

L114 · 5min 影响 · 11分钟前 · [🔍 异味](#) · [🟡 次要](#)

(二) 进行分析

1. 安全性分析

根据 SonarQube 分析结果，项目中存在安全性问题，这些问题涉及到数据传输、用户输入验证、权限控制等方面的漏洞，提示存在潜在的安全风险。

2. 可靠性分析

项目中存在大约 3800 个可靠性问题。这些问题包括未处理的异常、潜在的 bug 或错误的边界条件。如此大量的可靠性问题可能会导致系统发生崩溃。因此，必须针对这些问题逐一排查确保关键功能路径的稳定性，通过异常处理来降低系统故障的风险。

3. 可维护性分析

存在 1100 个可维护性问题。这些问题通常与代码的复杂性和重复性相关，导致维护成本增加和代码理解困难。为了提升代码的可维护性，应该重构过于复杂的函数或方法，减少深层嵌套，提取重复逻辑并进行模块化。

4. 测试覆盖率分析

项目当前覆盖率为 0.0%，应尽早引入自动化测试框架，编写单元测试并逐步提升代码覆盖率，尤其是核心功能部分，还能增强代码的可维护性和稳定性。

5. 代码重复性分析

项目的代码大约 13.3%的代码存在重复。这种重复的代码影响了代码的整洁性，还增加了维护的复杂度，后期应该识别并去除重复部分，提取出公共方法模块，提升代码的复用性和可维护性。

6. 安全热点分析

代码中有大量潜在的安全问题需要进一步审查，为避免潜在的安全漏洞，要针对这些安全热点进行深入分析，并采取安全防护措施。

7. 总结

SonarQube 的分析结果显示，项目在安全性、可靠性、可维护性、测试覆盖率和代码重复性方面尚欠缺。后期应该解决安全性和可靠性问题，重构复杂的代码，提高项目稳定性和维护性。为系统的长期的健康发展做考虑，打造一款完美的软件。

十一、总结与思考

（一）本次实践的收获与经验

通过本次实践，我在开源软件分析维护有了更深的理解和经验。通过对校园二手商城项目的学习，我有了更深的理解关于 Django 框架、DRF，还有前端 Vue 和微信小程序的集成开发。在实际操作中，我掌握了项目的部署、设置开发环境、配置 MySQL 和 Redis、使用 uwsgi 和 nginx 进行部署等。

通过代码分析工具 SonarQube 代码质量分析，使得我更意识到代码质量的重要性，尤其是在安全性和测试覆盖和可靠性和可维护性率等方面。

（二）评价开源软件的质量水平

校园二手商城项目是一个二次开发项目，功能较为完善，能够为校园用户的二手商品交易平台，满足了学生发布商品和求购信息一些日常的基本需求，同时集成小程序端与后端的统一交互，整体效果较好。

项目的后端使用了 Django 和 DRF，前端使用了 Vue，技术栈比较全面新颖。然而 SonarQube 的分析结果并不乐观，该项目还有许多方面的问题，后期还需要对项目进行优化提升。

（三）实验过程的优化提升

实验过程中，配置开发环境时遇到了一些困难，尤其是在 uwsgi 和 nginx 的配置上。

虽然有相关的部署文档，但一些具体操作的描述不够详细，特别是遇到错误时的解决方案。可以在操作过程中的常见问题记录下来，并在文档中提供详细的步骤，供其他同学共同解决使用。

代码分析工具 SonarQube 进行代码分析时，虽然能够得出最终的质量报告，但如何理解报告也是一个考验，应对这些报告需要更深入的了解，要根据分析结果进一步优化项目代码。

十二、参考文献

1. https://blog.csdn.net/hhyyqq/article/details/141282529?ops_request_misc=%257B%2522request%255Fid%2522%253A%25229f5b73d6a091bf7856f8ace531132219%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=9f5b73d6a091bf7856f8ace531132219&biz_id=0&utm_medium=distribute.pc_chrome_plugin_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v3l_ecpm-4-141282529-null-null.nonecase&utm_term=win%E4%B8%8B%E8%BD%BDuwsig&spm=1018.2226.3001.4187

2. <https://www.piis.cn/zhishi/2453.html>

3. [https://blog.csdn.net/weixin_42292991/article/details/95976016/?ops_request_misc=&request_id=&biz_id=102&utm_term=chdir\(\):%20No%20such%20file%20or%20direc&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-95976016.142^v102^pc_search_result_base3&spm=1018.2226.3001.4187](https://blog.csdn.net/weixin_42292991/article/details/95976016/?ops_request_misc=&request_id=&biz_id=102&utm_term=chdir():%20No%20such%20file%20or%20direc&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-95976016.142^v102^pc_search_result_base3&spm=1018.2226.3001.4187)

4. <https://github.com/3BPM/shshop>

5. https://blog.csdn.net/lmdsoft/article/details/82585126?ops_request_misc=&request_id=&biz_id=102&utm_term=sonar%20scanner%E6%89%AB%E6%8F%8F%E7%BB%93%E6%9E%9C%E5%88%86%E6%9E%90&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-9-82585126.nonecase&spm=1018.2226.3001.4187

6. https://blog.csdn.net/intelrain/article/details/95043570?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522027523848925426d4aadd82b6e40eala%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=027523848925426d4aadd82b6e40eala&biz_id=0&utm_medium=distribute.pc_chrome_plugin_search_result.none-task-blog-2~all~sobaiduweb~default-2-95043570-null-null.nonecase&utm_term=sonar%E6%89%AB%E6%8F%8F%E5%A4%B1%E8%B4%A5&spm=1018.2226.3001.4187

7. https://blog.csdn.net/Hatsune__Miku__/article/details/132866003

8. <https://blog.csdn.net/bfqs1988/article/details/80236753>

9. https://blog.csdn.net/weixin_48704094/article/details/132224975?ops_request_misc=&request_id=&biz_id=102&utm_term=sonarqube%E9%85%8D%E7%BD%AE%E7%8E%AF%E5%A2%83%E5%8F%98%E9%87%8Fjre&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-1-132224975.142^v102^pc_search_result_base3&spm=1018.2226.3001.4187

10. <https://so.csdn.net/so/search?q=sonarqube%E9%85%8D%E7%BD%AE%E7%8E%AF%E5%A2%83%E5%8F%98%E9%87%8Fjre&urw=>

11. https://so.csdn.net/so/search?from=chrome_plugin&q=Sonarqube%E8%BF%9E%E6%8E%A5%E6%95%B0%E6%8D%AE%E5%BA%93%E5%90%8E%E6%8A%A5%E9%94%99

12. https://blog.csdn.net/qq_41631176/article/details/84952820?spm=1001.2101.3001.6661.1&utm_medium=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7ECTRLIST%7EPaidSort-1-84952820-blog-90613736.235%5Ev43%5Epc_blog_bottom_relevance_base7&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7ECTRLIST%7EPaidSort-1-84952820-blog-90613736.235%5Ev43%5Epc_blog_bottom_relevance_base7&utm_relevant_index=1