温州大學甌江學院
WENZHOU UNIVERSITY OUJIANG COLLEGE

《爬虫综合实验》

题　　目：爬虫实验合集

分　　院：理工分院

班　　级：１６计算机科学与技术本３

姓　　名：王家跃

学　　号：16219111301

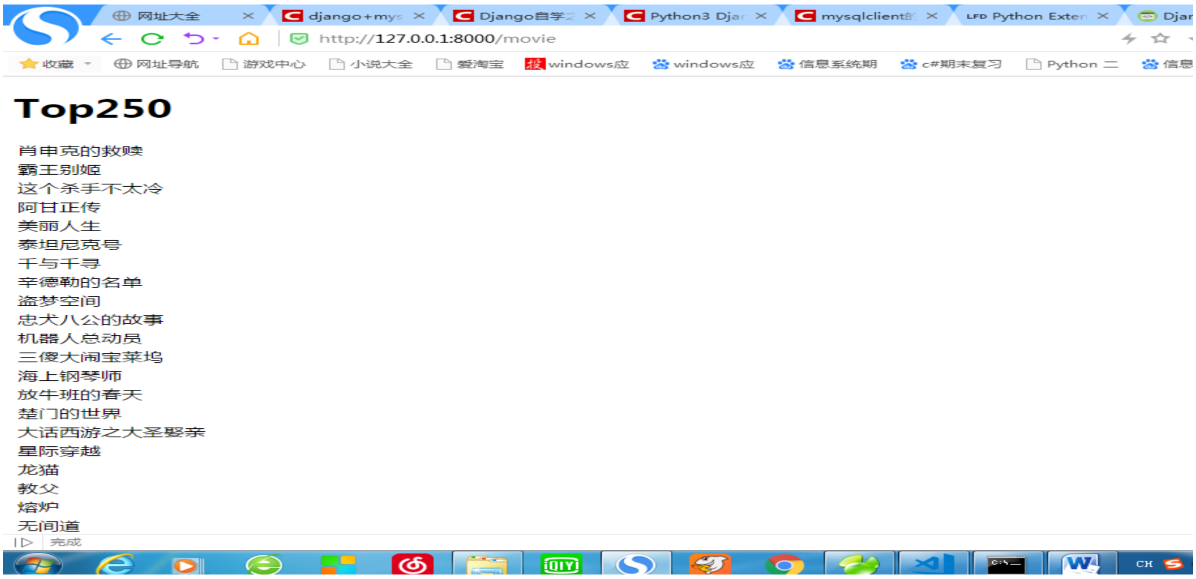
完成日期：2019 年 6 月 18 日

温州大学瓯江学院教务部

二〇一二年十一月制

一、期中作业

豆瓣最终结果图：

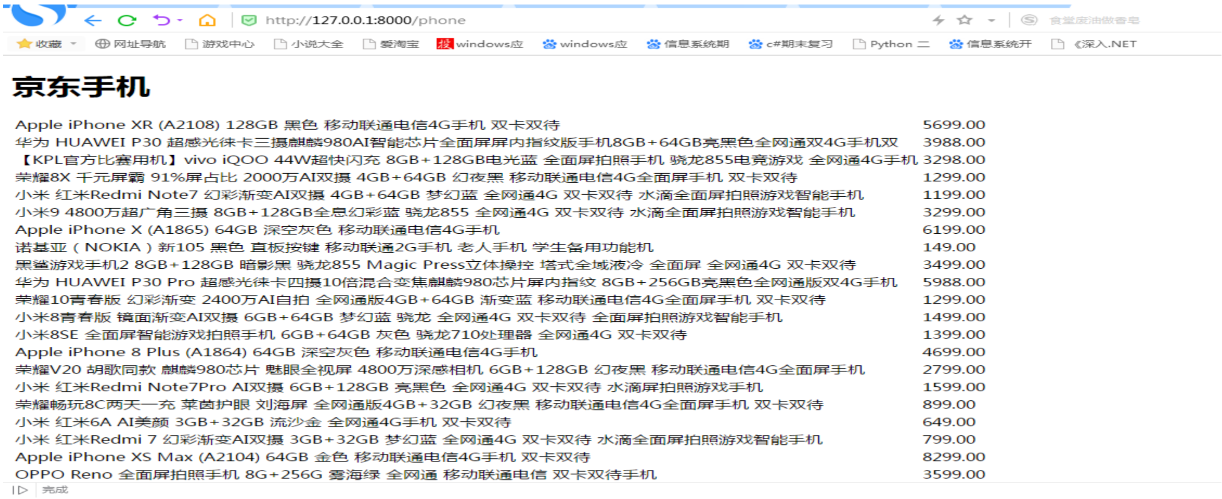


Top250

肖申克的救赎
霸王别姬
这个杀手不太冷
阿甘正传
美丽人生
泰坦尼克号
千与千寻
辛德勒的名单
盗梦空间
忠犬八公的故事
机器人总动员
三傻大闹宝莱坞
海上钢琴师
放牛班的春天
楚门的世界
大话西游之大圣娶亲
星际穿越
龙猫
教父
熔炉
无间道

爬取京东手机 Django 截图：

（只爬取 2 页的数据 ）



京东手机

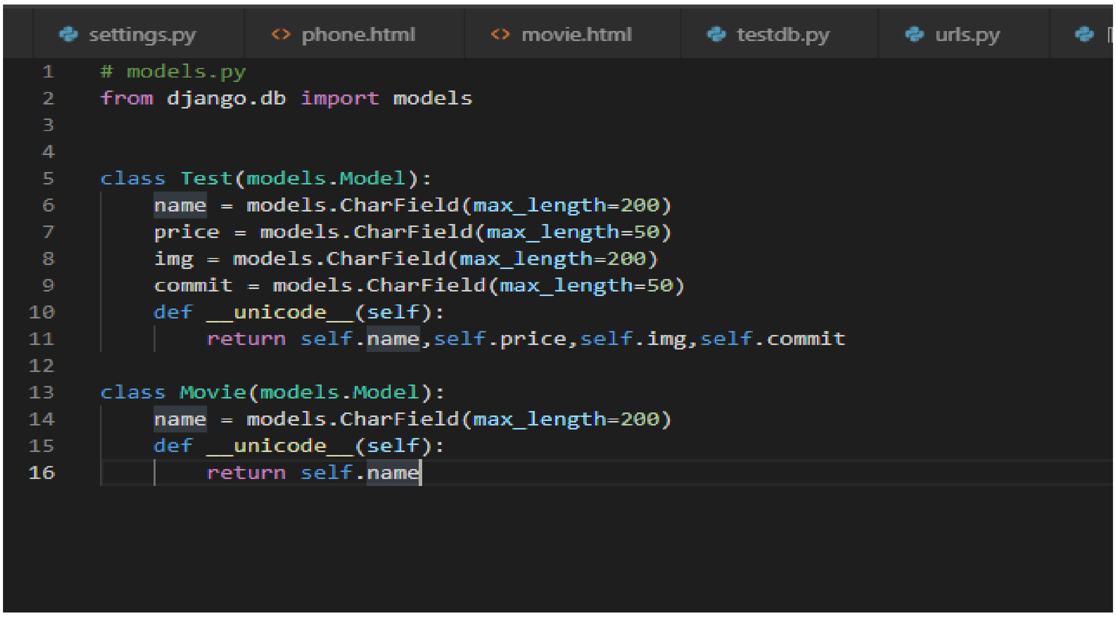
Apple iPhone XR (A2108) 128GB 黑色 移动联通电信4G手机 双卡双待	5699.00
华为 HUAWEI P30 超感光徕卡三摄麒麟980AI智能芯片全面屏屏内指纹版手机8GB+64GB亮黑色全网通双4G手机双	3988.00
【KPL官方比赛用机】vivo iQOO 44W超快闪充 8GB+128GB电光蓝 全面屏拍照手机 骁龙855电竞游戏 全网通4G手机	3298.00
荣耀8X 千元屏霸 91%屏占比 2000万AI双摄 4GB+64GB 幻夜黑 移动联通电信4G全面屏手机 双卡双待	1299.00
小米 红米Redmi Note7 幻彩渐变AI双摄 4GB+64GB 梦幻蓝 全网通4G 双卡双待 水滴全面屏拍照游戏智能手机	1199.00
小米9 4800万超广角三摄 8GB+128GB全息幻彩蓝 骁龙855 全网通4G 双卡双待 水滴全面屏拍照游戏智能手机	3299.00
Apple iPhone X (A1865) 64GB 深空灰色 移动联通电信4G手机	6199.00
诺基亚（ NOKIA ）新105 黑色 直板按键 移动联通2G手机 老人手机 学生备用功能机	149.00
黑鲨游戏手机2 8GB+128GB 暗影黑 骁龙855 Magic Press立体操控 塔式全域液冷 全面屏 全网通4G 双卡双待	3499.00
华为 HUAWEI P30 Pro 超感光徕卡四摄10倍混合变焦麒麟980芯片屏内指纹 8GB+256GB亮黑色全网通版双4G手机	5988.00
荣耀10青春版 幻彩渐变 2400万AI自拍 全网通版4GB+64GB 渐变蓝 移动联通电信4G全面屏手机 双卡双待	1299.00
小米8青春版 镜面渐变AI双摄 6GB+64GB 梦幻蓝 骁龙 全网通4G 双卡双待 全面屏拍照游戏智能手机	1499.00
小米8SE 全面屏智能游戏拍照手机 6GB+64GB 灰色 骁龙710处理器 全网通4G 双卡双待	1399.00
Apple iPhone 8 Plus (A1864) 64GB 深空灰色 移动联通电信4G手机	4699.00
荣耀V20 胡歌同款 麒麟980芯片 魅眼全视屏 4800万深感相机 6GB+128GB 幻夜黑 移动联通电信4G全面屏手机	2799.00
小米 红米Redmi Note7Pro AI双摄 6GB+128GB 亮黑色 全网通4G 双卡双待 水滴屏拍照游戏手机	1599.00
荣耀畅玩8C两天一充 莱茵护眼 刘海屏 全网通版4GB+32GB 幻夜黑 移动联通电信4G全面屏手机 双卡双待	899.00
小米 红米6A AI美颜 3GB+32GB 流沙金 全网通4G手机 双卡双待	649.00
小米 红米Redmi 7 幻彩渐变AI双摄 3GB+32GB 梦幻蓝 全网通4G 双卡双待 水滴全面屏拍照游戏智能手机	799.00
Apple iPhone XS Max (A2104) 64GB 金色 移动联通电信4G手机 双卡双待	8299.00
OPPO Reno 全面屏拍照手机 8G+256G 雾海绿 全网通 移动联通电信 双卡双待手机	3599.00

Django （爬取豆瓣 TOP250 和京东手机的价格）

1. 安装 django　pip install Django
2. 创建第一个项目：django-admin　startproject　JDdjango
3. 启动服务器：python manage.py runserver 0.0.0.0:8000
4. 新建 view.py
5. 数据库配置：

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'mypachou', #数据库名字
        'USER': 'root1',
        'PASSWORD': 'wangjiayue',
        'HOST':'localhost',
        'PORT':'3306',
    }
}
```

6. 创建一个 TestModel 的 APP: Django-admin.py startapp TestModel
7. TestModel/model.py 代码：建立一个 movie 表和一个 test 表（京东手机）



```
# models.py
from django.db import models


class Test(models.Model):
    name = models.CharField(max_length=200)
    price = models.CharField(max_length=50)
    img = models.CharField(max_length=200)
    commit = models.CharField(max_length=50)
    def __unicode__(self):
        return self.name,self.price,self.img,self.commit

class Movie(models.Model):
    name = models.CharField(max_length=200)
    def __unicode__(self):
        return self.name
```

8. 在 setting.py 中修改代码：

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'TestModel', #添加此项
]
```

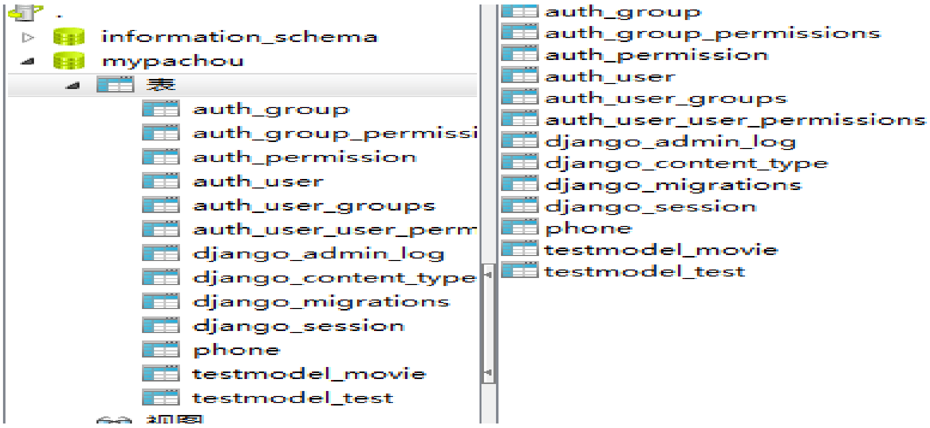
9. Cmd 运行：

Python manage.py migrate

Python manage.py makemigrations TestModel

Python manage.py migrate TestModel

三句运行成功后在 test 数据库下会生成 testmodel_test ,testmodel_movie 表



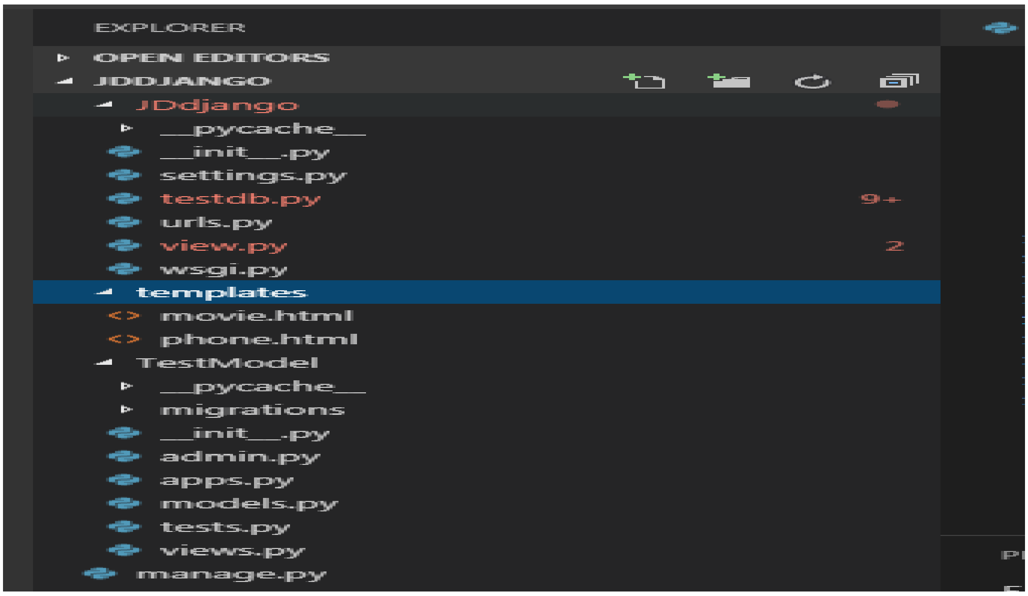
```
.
▷ information_schema
◢ mypachou
  ◢ 表
      auth_group
      auth_group_permissi
      auth_permission
      auth_user
      auth_user_groups
      auth_user_user_perm
      django_admin_log
      django_content_type
      django_migrations
      django_session
      phone
      testmodel_movie
      testmodel_test
  视图
```

```
auth_group
auth_group_permissions
auth_permission
auth_user
auth_user_groups
auth_user_user_permissions
django_admin_log
django_content_type
django_migrations
django_session
phone
testmodel_movie
testmodel_test
```

10. 在添加 testdb.py 代码来获取爬虫写入数据库的数据：

代码见源码

11. 目录：（导入 models 中的 Test 和 Movie 表，一直报错，不存在 objects 成员，没解决但是能实现）

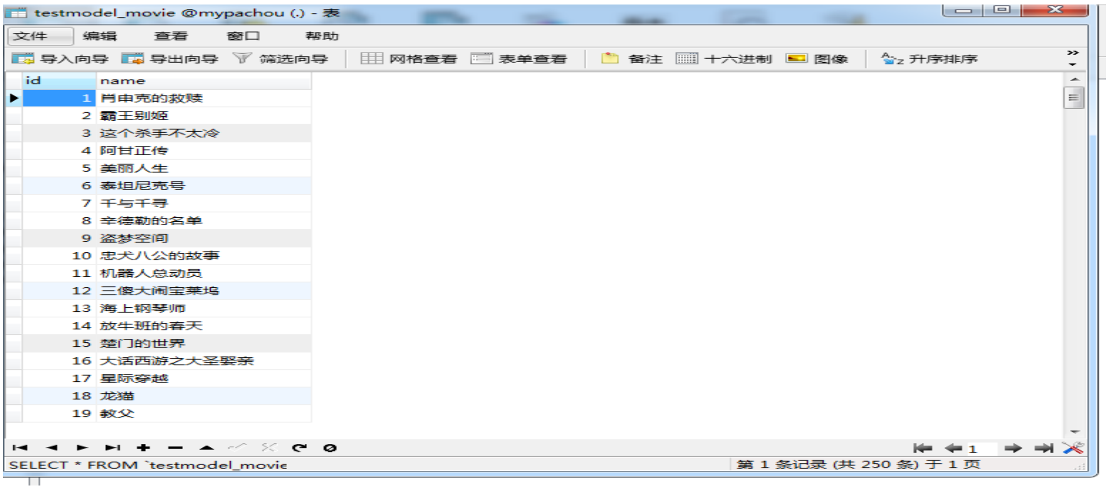


```
EXPLORER
▸ OPEN EDITORS
◢ JDDJANGO
  ◢ JDdjango
    ▸ __pycache__
      __init__.py
      settings.py
      testdb.py                9+
      urls.py
      view.py                  2
      wsgi.py
  ◢ templates
      movie.html
      phone.html
  ◢ TestModel
    ▸ __pycache__
    ▸ migrations
      __init__.py
      admin.py
      apps.py
      models.py
      tests.py
      views.py
    manage.py
```

在 python 分别运行爬取豆瓣和爬取京东手机的代码，将爬取的数据插入表中：

代码见源码

结果截图：



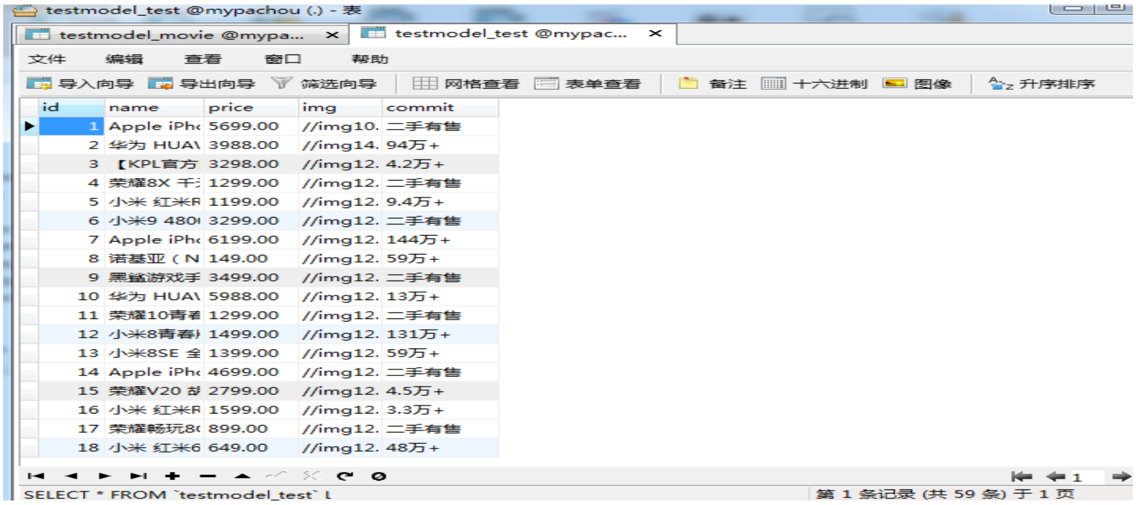
id	name
1	肖申克的救赎
2	霸王别姬
3	这个杀手不太冷
4	阿甘正传
5	美丽人生
6	泰坦尼克号
7	千与千寻
8	辛德勒的名单
9	盗梦空间
10	忠犬八公的故事
11	机器人总动员
12	三傻大闹宝莱坞
13	海上钢琴师
14	放牛班的春天
15	楚门的世界
16	大话西游之大圣娶亲
17	星际穿越
18	龙猫
19	教父

SELECT * FROM `testmodel_movie

第 1 条记录 (共 250 条) 于 1 页

（京东手机只爬取了两页内容，代码以实现全部爬取：）

截图：

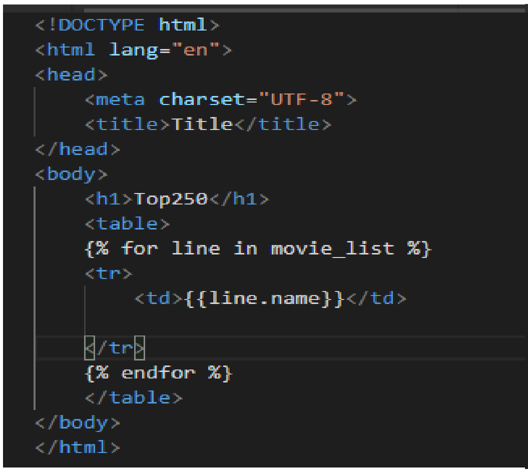


id	name	price	img	commit
1	Apple iPh	5699.00	//img10.	二手有售
2	华为 HUAV	3988.00	//img14.	94万+
3	【KPL官方	3298.00	//img12.	4.2万+
4	荣耀8X 千	1299.00	//img12.	二手有售
5	小米 红米R	1199.00	//img12.	9.4万+
6	小米9 480	3299.00	//img12.	二手有售
7	Apple iPh	6199.00	//img12.	144万+
8	诺基亚（N	149.00	//img12.	59万+
9	黑鲨游戏手	3499.00	//img12.	二手有售
10	华为 HUAV	5988.00	//img12.	13万+
11	荣耀10青者	1299.00	//img12.	二手有售
12	小米8青春	1499.00	//img12.	131万+
13	小米8SE 全	1399.00	//img12.	59万+
14	Apple iPh	4699.00	//img12.	二手有售
15	荣耀V20 甜	2799.00	//img12.	4.5万+
16	小米 红米R	1599.00	//img12.	3.3万+
17	荣耀畅玩8	899.00	//img12.	二手有售
18	小米 红米6	649.00	//img12.	48万+

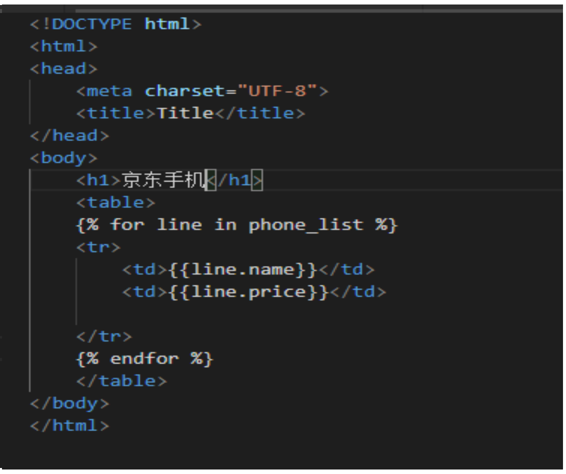
SELECT * FROM `testmodel_test` L

第 1 条记录 (共 59 条) 于 1 页

新建 templates 文件夹并在文件夹下新建 phone.html movie.html:

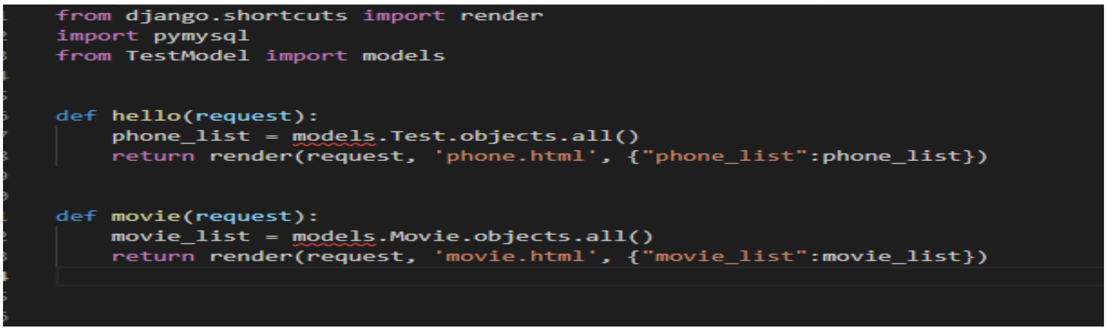


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
    <h1>Top250</h1>
    <table>
    {% for line in movie_list %}
    <tr>
        <td>{{line.name}}</td>

    </tr>
    {% endfor %}
    </table>
</body>
</html>
```



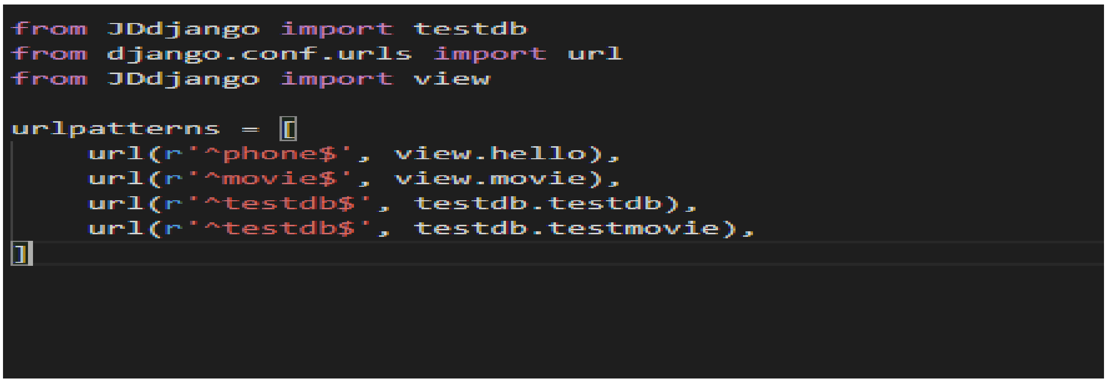
```html
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
    <h1>京东手机</h1>
    <table>
    {% for line in phone_list %}
    <tr>
        <td>{{line.name}}</td>
        <td>{{line.price}}</td>

    </tr>
    {% endfor %}
    </table>
</body>
</html>
```

修改 View.py：



```
from django.shortcuts import render
import pymysql
from TestModel import models


def hello(request):
    phone_list = models.Test.objects.all()
    return render(request, 'phone.html', {"phone_list":phone_list})


def movie(request):
    movie_list = models.Movie.objects.all()
    return render(request, 'movie.html', {"movie_list":movie_list})
```

以及绑定到 urls.py:



```
from JDdjango import testdb
from django.conf.urls import url
from JDdjango import view

urlpatterns = [
    url(r'^phone$', view.hello),
    url(r'^movie$', view.movie),
    url(r'^testdb$', testdb.testdb),
    url(r'^testdb$', testdb.testmovie),
]
```

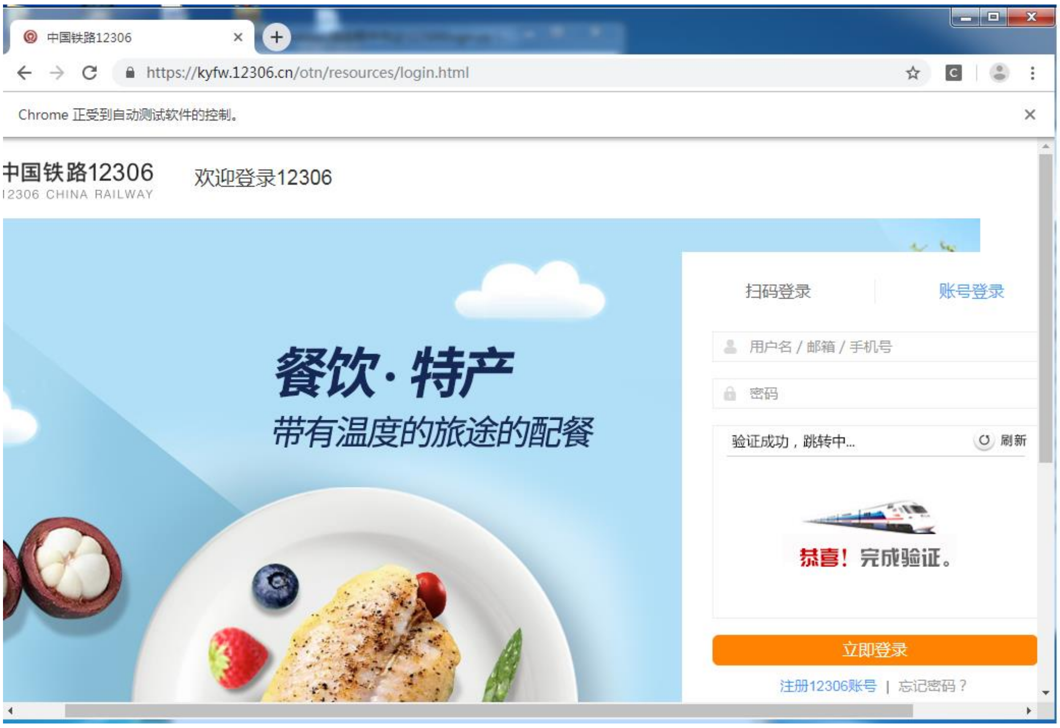
二、期末作业

12306 自动登陆代码（基本实现全自动化）:

```
import re
import time
import base64
import requests
import sys
from selenium import webdriver
from selenium.webdriver.support.ui import WebDriverWait
import urllib.request
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.action_chains import ActionChains

class Login(object):
    def __init__(self):
        self.login_url = "https://kyfw.12306.cn/otn/resources/login.html"
        self.geturl = "http://littlebigluo.qicp.net:47720/"
        self.totalFlush = 0
        self.startTime = time.time()
        # self.driver = '' #驱动 chrome 浏览器进行操作
        driver = webdriver.Chrome()
        self.driver = driver #驱动 chrome 浏览器进行操作


    def login_input(self):
        self.driver.get(self.login_url)
        time.sleep(0.2)
        account = self.driver.find_element_by_class_name("login-hd-account")
        account.click()
        userName = self.driver.find_element_by_id("J-userName")
        userName.send_keys("15606879517")    # 12306 账号
        password = self.driver.find_element_by_id("J-password")
        password.send_keys("64890110wjy")    # 12306 密码
        time.sleep(5)


    def getImage(self):
        img_element =WebDriverWait(self.driver,
100).until(EC.presence_of_element_located((By.ID, "J-loginImg")))
        try:
            img_element
        except Exception as e:
            print("出错了，请稍后再试")
        base64_str=img_element.get_attribute("src").split(",")[-1]
        imgdata=base64.b64decode(base64_str)
        with open('mylogin.jpg','wb') as file:
            file.write(imgdata)
        self.img_element=img_element
        time.sleep(2)


    def getResult(self):
        driver = webdriver.Chrome()
        driver.get(self.geturl)
        time.sleep(5)
        files ="C:\\Users\\Administrator\\Desktop\\爬虫期中作业\\mylogin.jpg"
        onput = driver.find_element_by_name("pic_xxfile")
        onput.send_keys(files)
        myclick = driver.find_element_by_xpath("/html/body/form/input[2]").click()
        result = driver.find_element_by_xpath("/html/body/p[1]/font/font")
        result=result.text.split(" ")
        print(result)
        self.result=result
        time.sleep(2)
        driver.quit()



    def Click(self):
        js=""
        for i in self.result:
            if i=='1':
                js=js+'<div randcode="42,50" class="lgcode-active" style="top: 66px; left: 29px;"></div>'
            elif i=='2':
                js=js+'<div randcode="109,43" class="lgcode-active" style="top: 59px; left: 96px;"></div>'
            elif i=='3':
                js=js+'<div randcode="186,36" class="lgcode-active" style="top: 52px; left: 178px;"></div>'
            elif i=='4':
                js=js+'<div randcode="256,50" class="lgcode-active" style="top: 62px; left: 241px;"></div>'
            elif i=='5':
                js=js+'<div randcode="31,124" class="lgcode-active" style="top: 140px; left: 18px"></div>'
            elif i=='6':
                js=js+'<div randcode="119,110" class="lgcode-active" style="top: 130px; left: 103px;"></div>'
            elif i=='7':
                js=js+'<div randcode="182,110" class="lgcode-active" style="top: 124px; left: 169px;"></div>'
            else :
                js=js+'<div randcode="258,110" class="lgcode-active" style="top: 128px; left: 245px;"></div>'
        resultjs='document.getElementById("J-passCodeCoin").innerHTML='+"'"+js+"'"
        print(resultjs)
        self.driver.execute_script(resultjs)
        time.sleep(5)




    def submit(self):
        self.driver.find_element_by_id("J-login").click()
```

```
if __name__ == '__main__':
        spider = Login()
        spider.login_input()
        spider.getImage()
        spider.getResult()
        spider.Click()
        spider.submit()
```

成功截图：



https://kyfw.12306.cn/otn/resources/login.html

Chrome 正受到自动测试软件的控制。

验证成功，跳转中...

恭喜！完成验证。