APPENDIX

APPENDIX A

Crawler Program code for JD

Crawler Program code for JD

Items.py

```
import scrapy


class JdspiderItem(scrapy.Item):
    # define the fields for your item here like:
    # name = scrapy.Field()
    pass

class JDCommentItem(scrapy.Item):
    productId = scrapy.Field()
    id = scrapy.Field()
    nickname = scrapy.Field()
    score = scrapy.Field()
    comment = scrapy.Field()
    creationTime = scrapy.Field()
    productColor = scrapy.Field()
    productSize = scrapy.Field()
    image_num = scrapy.Field()
    video_num = scrapy.Field()
```

middlewares.py

```
from scrapy import signals

# useful for handling different item types with a single interface
from itemadapter import is_item, ItemAdapter

from selenium import webdriver
from scrapy.http import HtmlResponse

class JdspiderSpiderMiddleware:
    # Not all methods need to be defined. If a method is not defined,
    # scrapy acts as if the spider middleware does not modify the
    # passed objects.

    @classmethod
    def from_crawler(cls, crawler):
        # This method is used by Scrapy to create your spiders.
```

```
        s = cls()
        crawler.signals.connect(s.spider_opened, signal=signals.spider_opened)
        return s

    def process_spider_input(self, response, spider):
        # Called for each response that goes through the spider
        # middleware and into the spider.

        # Should return None or raise an exception.
        return None

    def process_spider_output(self, response, result, spider):
        # Called with the results returned from the Spider, after
        # it has processed the response.

        # Must return an iterable of Request, or item objects.
        for i in result:
            yield i

    def process_spider_exception(self, response, exception, spider):
        # Called when a spider or process_spider_input() method
        # (from other spider middleware) raises an exception.

        # Should return either None or an iterable of Request or item objects.
        pass

    def process_start_requests(self, start_requests, spider):
        # Called with the start requests of the spider, and works
        # similarly to the process_spider_output() method, except
        # that it doesn't have a response associated.

        # Must return only requests (not items).

        for r in start_requests:
            yield r

    def spider_opened(self, spider):
        spider.logger.info('Spider opened: %s' % spider.name)


class JdspiderDownloaderMiddleware:
    # Not all methods need to be defined. If a method is not defined,
```

```
# scrapy acts as if the downloader middleware does not modify the
# passed objects.

@classmethod
def from_crawler(cls, crawler):
    # This method is used by Scrapy to create your spiders.
    s = cls()
    crawler.signals.connect(s.spider_opened, signal=signals.spider_opened)
    return s

def __init__(self):
    self.driver = webdriver.Edge()
def __del__(self):
    self.driver.close()

def process_request(self, request, spider):
    # Called for each request that goes through the downloader
    # middleware.

    # Must either:
    # - return None: continue processing this request
    # - or return a Response object
    # - or return a Request object
    # - or raise IgnoreRequest: process_exception() methods of
    #   installed downloader middleware will be called



    self.driver.get(request.url)

    response = HtmlResponse(url=request.url, body=self.driver.page_source,
                            request=request, encoding='utf-8')

    return response

def process_response(self, request, response, spider):
    # Called with the response returned from the downloader.

    # Must either;
    # - return a Response object
    # - return a Request object
    # - or raise IgnoreRequest
```

```
        return response

    def process_exception(self, request, exception, spider):
        # Called when a download handler or a process_request()
        # (from other downloader middleware) raises an exception.

        # Must either:
        # - return None: continue processing this exception
        # - return a Response object: stops process_exception() chain
        # - return a Request object: stops process_exception() chain
        pass

    def spider_opened(self, spider):
        spider.logger.info('Spider opened: %s' % spider.name)
```

pipelines.py

```
import csv


from itemadapter import ItemAdapter
#import pymysql


# class DbPipeline:
#     def __init__(self):
#         #建立链接
#         self.conn = pymysql.connect(host='localhost',
#                                     port=3306,
#                                     user='root',
#                                     password='1234',
#                                     database='spider',
#                                     charset='utf8mb4')
#         #创建游标
#         self.cursor = self.conn.cursor()
#         self.data = []
#
#     def close_spider(self,spider):
#         if len(self.data) > 0:
#             #保存到数据库
#             self._write_to_db()
```

```
#       #爬虫程序关闭的时候关闭
#
#       self.conn.close()
#    def process_item(self, item, spider):
#       #每拿到一条数据都会调用
#
#       self.data.append((item['productId'], item['nickname'], item['score'], item['comment'], item['productColor']))
#       #每 100 条保存一下
#       if len(self.data) == 100:
#           #保存到数据库
#           self._write_to_db()
#           self.data.clear()
#       return item

    # def _write_to_db(self):
    #     self.cursor.executemany(
    #         'insert into huawei_mate_50 (uid, nickname, score, comment, productColor)'
    #         ' values (%s, %s, %s, %s, %s)',
    #         self.data
    #     )
    #     # 提交然后清空容器
    #     self.conn.commit()

from JDSpider.spiders.JDSprider import name_id
import openpyxl
import os

from datetime import datetime
#创建管道保存数据到 excel
class ExcelPipeline:

    # def __init__(self):
    #     #创建工作簿
    #
    #     self.num = 1
    #     self.target_time = '2023-03-10 11:38:53'
    #     self.format_pattern = '%Y-%m-%d %H:%M:%S'
    #     self.filename = "京东化妆品评论.xlsx"
    #     if os.path.isfile(self.filename):
    #         self.wb = openpyxl.load_workbook(self.filename)
    #         self.ws = self.wb.active
```

```
    #     else:
    #         self.wb = openpyxl.Workbook()
    #         self.ws = self.wb.active
    #         self.ws.append(['产品 Id', '用户名 id', '用户名', '评分', '评论', '评论时间', '颜色', '内存大小'])
    #

    def open_spider(self,spider):
        #爬虫开始时候调用
        pass
    def close_spider(self,spider):
        #爬虫程序关闭的时候保存
        pass
        #self.wb.save(self.filename)
    def process_item(self, item, spider):
        #每拿到一条数据都会调用

        # difference = (datetime.strptime(item['creationTime'], self.format_pattern) -
datetime.strptime(self.target_time,self.format_pattern))
        # #if difference.days > 0:
        # if True:
        #     self.num += 1
        #     self.ws.append((item['productId'], item['id'],item['nickname'], item['score'], item['comment'],
item['creationTime'],item['productColor'],item['productSize']))#取到空值会报错
        #
        #     if self.num == 10:
        #         #保存到数据库
        #         self.wb.save(self.filename)
        #         self.num = 1

        ccc = "产品 Id: {}, 用户名 id: {}, 用户名：{}, 评分：{},评论：{},评论时间:{},颜色：{},大小：{},图片数：{},视频数：{}".format(item['productId'], item['id'],item['nickname'], item['score'],
item['comment'], item['creationTime'],item['productColor'],item['productSize'],item['image_num'],item['video_num'])
+ "\n"
        with open("jingdong1.txt", "a", encoding='utf-8') as f:
            f.write(ccc)


        list = [item['productId'], item['id'],item['nickname'], item['score'], item['comment'],
item['creationTime'],item['productColor'],item['productSize'],item['image_num'],item['video_num']]
        with open("jingdong1.csv", "a", encoding='utf-8', newline="") as f:
```

```
        k = csv.writer(f, dialect="excel")
        with open("jingdong1.csv", "r", encoding='utf-8', newline="") as f:
            reader = csv.reader(f)
            if not [row for row in reader]:
                k.writerow(['产品 Id', '用户名 id', '用户名', '评分', '评论', '评论时间', '颜色', '大小','
图片数', '视频数'])
                k.writerow(list)
            else:
                k.writerow(list)

    return item
```

APPENDIX

APPENDIX B

Crawler Program code for Xiaohongshu

Crawler Program code for Xiaohongshu

```
# -*- encoding=utf8 -*-
__author__ = "Alan"
from tkinter import *
import csv
import random
from datetime import datetime
import re
import time

from airtest.core.api import *

auto_setup(__file__)

# -*- encoding=utf8 -*-
__author__ = "Alan"

import time

from airtest.core.api import *



from poco.drivers.android.uiautomation import AndroidUiautomationPoco

# 连接本机默认端口连的一台设备号为 SJE5T17B17 的手机
# auto_setup(__file__,devices=["Android://127.0.0.1:5037/S2D0218A10003035"])

auto_setup(__file__)
# set_current(0)
# #初始化第 1 台设备
poco=AndroidUiautomationPoco(use_airtest_input=True, screenshot_each_action=False)



root= Tk()
root.title('爬虫')
root.geometry('540x440') # 这里的乘号不是 * ， 而是小写英文字母 x
```

```
text = Text(root, height=20, width=50)
text.pack()

entry = Entry(root,width=20)
entry.pack()

var = IntVar()  # 保存为一个 int 类型的变量
var.set(0)  # 设置初始值
Label(root, text="获取到", font=("黑体", 14), fg="red", width=12, height=2).place(x=100, y=350, anchor='nw')


def rungra(key_words,tar_time,poco):
    #亮屏
    wake()
    #点击 home 键
    home()
#    打开小红书
    start_app('com.xingin.xhs',activity=None)
    sleep(1)
    stop_app('com.xingin.xhs')

    sleep(2)

    start_app('com.xingin.xhs',activity=None)

    #等待+号出现
    poco(name="com.xingin.xhs:id/cqs").wait(15)
    sleep(2)
    #搜索
    poco(name="com.xingin.xhs:id/g1c").click()
    sleep(2)

    for word in key_words:
        # 搜索关键字
        poco(name="com.xingin.xhs:id/e1l").set_text(word)
        time.sleep(1)
        # 点击搜索
```

```
        poco(name="com.xingin.xhs:id/e1q").click()
        sleep(1.0)


        poco(text="全部 bitmap").click()

        poco(text="最新").click()


        # 判断网络
        isLast = poco(name="com.xingin.xhs:id/dtg").exists()
        if isLast:
            if (poco(name="com.xingin.xhs:id/dtg").get_text() == '网络好像断了，请检查手机是否联网
'):
                swipe = False
                poco.swipe([0.5, 0.2], [0.5, 0.5])


        # 如果还在笔记里点返回
        while poco("com.xingin.xhs:id/nickNameTV").exists():
            poco("com.xingin.xhs:id/rr").click()

            # 是否在外面页面
            if poco("android.widget.FrameLayout").child("android.widget.LinearLayout").offspring(
                    "android:id/content").offspring("com.xingin.xhs:id/e16").offspring("com.xingin.xhs:id/e15").child(
                "android.widget.FrameLayout").exists():
                break

        # 如果还在视频里点返回
        while poco("com.xingin.xhs:id/matrixNickNameView").exists():
            poco("com.xingin.xhs:id/backButton").click()

            # 是否在外面页面
            if poco("android.widget.FrameLayout").child("android.widget.LinearLayout").offspring(
                    "android:id/content").offspring("com.xingin.xhs:id/e16").offspring("com.xingin.xhs:id/e15").child(
                "android.widget.FrameLayout").exists():
                break
```

```
notes = []
num = 0
swipe = True
while swipe:
    # 笔记列表
    laouts = poco("android.widget.FrameLayout").child("android.widget.LinearLayout").offspring(
        "android:id/content").offspring("com.xingin.xhs:id/e16").offspring("com.xingin.xhs:id/e15").child(
        "android.widget.FrameLayout")
    for laout in laouts:
        duration = random.uniform(0.9, 1.0)
        try:
            #标题和用户名是否存在
            if laout.offspring("com.xingin.xhs:id/esg").exists() and laout.offspring("com.xingin.xhs:id/pv").exists():
                title = laout.offspring("com.xingin.xhs:id/esg").get_text()
                user_name = laout.offspring("com.xingin.xhs:id/pv").get_text()
                one_note = [title, user_name]
                #判断当前的标题和用户名是否已经拿过
                if one_note not in notes:
                    # 标题和用户名保存到容器里，未来判断是否拿过
                    notes.append([title, user_name])

                    # 点击进入笔记详情
                    laout.offspring("com.xingin.xhs:id/esg").click()
                    #判断是否是视频
                    if
poco("android.widget.LinearLayout").offspring("com.xingin.xhs:id/hv4").offspring("android.view.View").exists():
                        notetype ="video"

                        if poco(name="com.xingin.xhs:id/e8o").exists():
                            love = poco(name="com.xingin.xhs:id/e8o").get_text()
                        else:
                            love=''
                        if poco(name="com.xingin.xhs:id/e8k").exists():
                            star = poco(name="com.xingin.xhs:id/e8k").get_text()
                        else:
                            star=''
                        if poco(name="com.xingin.xhs:id/e8m").exists():
                            comment = poco(name="com.xingin.xhs:id/e8m").get_text()
                        else:
```

```
                comment=''
                #点击文本

                if exists(Template(r"tpl1680262463081.png", record_pos=(0.209, 0.755), resolution=(1080,
2340))):
                    touch(Template(r"tpl1680262523568.png", record_pos=(0.165, 0.748), resolution=(1080,
2340)))
                elif exists(Template(r"tpl1680262523568.png", record_pos=(0.209, 0.755), resolution=(1080,
2340))):
                    touch(Template(r"tpl1680262523568.png", record_pos=(0.165, 0.748), resolution=(1080,
2340)))
                else:
                    poco(name="com.xingin.xhs:id/noteContentText").click()



                if poco(name="com.xingin.xhs:id/e7l").exists() and
poco(name="com.xingin.xhs:id/e7e").exists():
                    poco(name="com.xingin.xhs:id/e7a").click()

                #获取详情
                if poco(name="com.xingin.xhs:id/era").exists():
                    desc = poco(name="com.xingin.xhs:id/era").get_text()

                    while not poco(name="com.xingin.xhs:id/gxa").exists():
                    poco.swipe([0.5,0.7],[0.5,0.3])

                        #是否在外面
                        if poco("android.widget.FrameLayout").child("android.widget.LinearLayout").offspring(
    "android:id/content").offspring("com.xingin.xhs:id/e16").offspring(
"com.xingin.xhs:id/e15").child(
"android.widget.FrameLayout").exists():
                            break

                    edi_time = poco(name="com.xingin.xhs:id/gxa").get_text()
                    # 关闭
                    while poco(name="com.xingin.xhs:id/akw").exists():
                        poco(name="com.xingin.xhs:id/akw").click()

                elif poco(name="com.xingin.xhs:id/eqq").exists():
```

```
                desc = poco(name="com.xingin.xhs:id/eqq").get_text()
                edi_time = poco(name="com.xingin.xhs:id/gxa").get_text()
            else:
                desc = ""
                edi_time = poco(name="com.xingin.xhs:id/gxa").get_text()

            now = datetime.now()


            while poco(name="com.xingin.xhs:id/noteContentText").exists() or
poco(name="com.xingin.xhs:id/eqq").exists():
                keyevent("back")



        else:
            notetype = "note"
            # 是否在外面页面
            # if poco("android.widget.FrameLayout").child("android.widget.LinearLayout").offspring(
            #         "android:id/content").offspring("com.xingin.xhs:id/e16").offspring(
            #     "com.xingin.xhs:id/e15").child(
            #     "android.widget.FrameLayout").exists():
            #     break


            #滑动直到标题和内容其中一个出现，这里加内容是防止有些笔记没有标题会死循环
            while not poco(name="com.xingin.xhs:id/esh").exists() and not
poco(name="com.xingin.xhs:id/cny").exists():
                poco.swipe([0.5, 0.8], [0.5, 0.5], duration=duration)


                # 是否在外面页面
                if poco("android.widget.FrameLayout").child("android.widget.LinearLayout").offspring(
                        "android:id/content").offspring("com.xingin.xhs:id/e16").offspring(
                    "com.xingin.xhs:id/e15").child(
                    "android.widget.FrameLayout").exists():
                    break


            #标题
            if poco(name="com.xingin.xhs:id/esh").exists():
```

```
                title = poco(name="com.xingin.xhs:id/esh").get_text()
                #滑动直到详情或者时间出现
                while not poco(name="com.xingin.xhs:id/cny").exists() and not
poco(name="com.xingin.xhs:id/es3").exists():
                    #如果评论出现停止
                    if poco("com.xingin.xhs:id/er5").exists():
                        break
                    poco.swipe([0.5, 0.8], [0.5, 0.4], duration=duration)
                    # 是否在外面页面
                    if poco("android.widget.FrameLayout").child("android.widget.LinearLayout").offspring(
                            "android:id/content").offspring("com.xingin.xhs:id/e16").offspring(
                        "com.xingin.xhs:id/e15").child(
                        "android.widget.FrameLayout").exists():
                        break
                # 滑动直到编辑时间出现
                while not poco(name="com.xingin.xhs:id/es3").exists():

                    #如果评论出现停止
                    if poco("com.xingin.xhs:id/er5").exists():
                        break



                    if poco("android.widget.FrameLayout").child("android.widget.LinearLayout").offspring(
        "android:id/content").offspring("com.xingin.xhs:id/e16").offspring(
    "com.xingin.xhs:id/e15").child(
    "android.widget.FrameLayout").exists():
                        break


                    poco.swipe([0.5, 0.8], [0.5, 0.4], duration=duration)


        #                   #是否在外面页面
        #                   if poco("android.widget.FrameLayout").child("android.widget.LinearLayout").offspring(
        # "android:id/content").offspring("com.xingin.xhs:id/e16").offspring("com.xingin.xhs:id/e15").child(
        # "android.widget.FrameLayout").exists():
        #                       break


                    # # 是否在外面页面
                    # if poco("android.widget.FrameLayout").child(
```

```
#         "android.widget.LinearLayout").offspring(
#         "android:id/content").offspring("com.xingin.xhs:id/e16").offspring(
#     "com.xingin.xhs:id/e15").child(
#     "android.widget.FrameLayout").exists():
#     break
#编辑时间
edi_time = poco(name="com.xingin.xhs:id/es3").get_text()
#当前系统时间
now = datetime.now()
#如果有详情取值，没有则为空
if poco(name="com.xingin.xhs:id/cny").exists():
    desc = poco(name="com.xingin.xhs:id/cny").get_text()
else:
    desc = ""

if poco(name="com.xingin.xhs:id/ers").exists():
    love = poco(name="com.xingin.xhs:id/ers").get_text()
else:
    love=''

if poco(name="com.xingin.xhs:id/eqe").exists():
    star = poco(name="com.xingin.xhs:id/eqe").get_text()
else:
    star=''

if poco(name="com.xingin.xhs:id/eql").exists():
    comment = poco(name="com.xingin.xhs:id/eql").get_text()
else:
    comment=''



    poco(name="com.xingin.xhs:id/rr").click()

                    #去除换行符


desc = desc.replace('\n',' ')
desc = ''.join(e for e in desc if e.isalnum() or e in ['。','！','，','？','；','：','（','）','“','”'])
title = ''.join(e for e in title if e.isalnum() or e in ['。','！','，','？','；','：','（','）','“','”'])
```

```
                love = love if love != '点赞' else 0
                comment = comment if comment != '评论' else 0
                star = star if star != '收藏' else 0

                gra_time = now
                time_dec=edi_time.split(' ')
                lenth = len(time_dec)
                if lenth==1:
                    if len(time_dec[0].split('-'))==2:
                        edi_time = str(gra_time.year)+'-'+time_dec[0]
                    else:
                        edi_time = time_dec[0]
                elif lenth==2:
                    if time_dec[0] == '编辑于':
                        if len(time_dec[-1].split('-'))==2:
                            edi_time = str(gra_time.year)+'-'+time_dec[-1]
                        else:
                            edi_time = time_dec[-1]
                    else:
                        edi_time = str(gra_time.year)+'-'+time_dec[-2]
                elif lenth==3:
                    if time_dec[0]=='今天':
                        edi_time = str(gra_time.year)+'-'+str(gra_time.month)+'-'+str(gra_time.day)+' '+time_dec[-
2]
                    elif time_dec[0]=='昨天':
                        edi_time = str(gra_time.year)+'-'+str(gra_time.month)+'-'+str(gra_time.day-1)+'
'+time_dec[-2]
                    elif time_dec[0]=='编辑于':
                        edi_time = str(gra_time.year)+'-'+time_dec[1]
                elif lenth==4:
                    if time_dec[1]=='今天':
                        edi_time = str(gra_time.year)+'-'+str(gra_time.month)+'-'+str(gra_time.day)+' '+time_dec[-
2]
                    elif time_dec[1]=='昨天':
                        edi_time = str(gra_time.year)+'-'+str(gra_time.month)+'-'+str(gra_time.day-1)+'
'+time_dec[-2]


                try:
```

```
                str_time = edi_time.split(' ')[0]
                note_time = datetime.strptime(str_time,'%Y-%m-%d')
                #转换为时间戳
                note_mktime = int(time.mktime(note_time.timetuple()))

                if note_mktime < tar_time:
                    swipe=False
            except:
                note_time=now
            ccc = "produlD:{},时间: {},抓取时间: {}, 标题： {}, 详情： {},喜欢： {},评论： {},
加星： {},用户名： {}".format(
                word,edi_time, now, title, desc, love, comment,star,user_name,) + "\n"
            num=num+1
            print(num,ccc)
            #保存文本
            with open("小红书笔记 new.txt", "a", encoding='utf-8') as f:
                f.write(ccc)
            list = [word, edi_time, now, title, desc, love, comment,star,user_name,]
            #保存为 csv
            with open("小红书笔记 new.csv", "a", encoding='utf-8', newline="") as f:

                k = csv.writer(f, dialect="excel")
                with open("小红书笔记 new.csv", "r", encoding='utf-8', newline="") as f:
                    reader = csv.reader(f)
                    if not [row for row in reader]:
                        k.writerow(
                            ["产品名称", "时间", "抓取时间","标题", "详情", "喜欢", "评论", "加
星", "用户名"])
                        k.writerow(list)
                    else:
                        k.writerow(list)


            var.set(var.get() + 1)  # 变化的值，此处修改为你的变量
            Label(root, text=str(var.get()), font=("黑体", 14), fg="red", width=12, height=2).place(
                x=150, y=350, anchor='nw')
            root.update()  # 不断更新
```

```
    except:
        print("出错 laout： ",len(laouts))
    if len(notes)>10:
        notes.pop(0)

    #如果还在笔记里点返回
    while poco("com.xingin.xhs:id/nickNameTV").exists():
        poco("com.xingin.xhs:id/rr").click()

        # 是否在外面页面
        if poco("android.widget.FrameLayout").child("android.widget.LinearLayout").offspring(
                "android:id/content").offspring("com.xingin.xhs:id/e16").offspring("com.xingin.xhs:id/e15").child(
            "android.widget.FrameLayout").exists():
            break

    #如果还在视频里点返回
    while poco("com.xingin.xhs:id/matrixNickNameView").exists():
        poco("com.xingin.xhs:id/backButton").click()

        # 是否在外面页面
        if poco("android.widget.FrameLayout").child("android.widget.LinearLayout").offspring(
                "android:id/content").offspring("com.xingin.xhs:id/e16").offspring("com.xingin.xhs:id/e15").child(
            "android.widget.FrameLayout").exists():
            break
poco.swipe([0.5, 0.8], [0.5, 0.3], duration=duration)
#是否在加载
while poco(name="com.xingin.xhs:id/djv").exists():
    poco.swipe([0.5, 0.3], [0.5, 0.8], duration=duration)
    sleep(1)
    poco.swipe([0.5, 0.9], [0.5, 0.2], duration=duration)


if num >1000:
    swipe=False


# 判断是否已到底
isLast = poco(name="com.xingin.xhs:id/dc8").exists()
```

```
            if isLast:
                if (poco(name="com.xingin.xhs:id/dc8").get_text() == '无更多内容'):
                    swipe = False

            if swipe == False:
                poco.swipe([0.5, 0.2], [0.5, 0.5])
                poco(name="com.xingin.xhs:id/e1d").click()



with open("word.txt", "r") as f:  # 打开文件
    data = f.read()  # 读取文件
    #插入数据
    text.insert("1.0", data)

def getTextInput():
    result = text.get("1.0","end")
    word = result.split("\n")
    word = [i for i in word if i != '']

    with open("word.txt", "w") as f:
        f.write(result)

    tar_time = entry.get()

    tar_time = datetime.strptime(tar_time, '%Y-%m-%d')
    # 转换为时间戳
    tar_time = int(time.mktime(tar_time.timetuple()))

    rungra(word,tar_time,poco)

btn = Button(root,height=1,width=5,text="开始",command=getTextInput)
btn.pack()
root.mainloop()
```

Bibliography

Name: Liu Yang

Day Month Year of Birth: 10/12/1986

Address: Unit 2, Building 6, No. 1 Country Garden Academy, Huaxi District, Guiyang City, Guizhou Province

Education:

2005 - 2009	Bachelor of Management, E-commerce, School of Science and Technology, East China Jiaotong University, Nanchang, China
2021 - 2023	Master of Business Administration (MBA), Business Administration, Dhonburi Rajabhat University, Thailand.

Position and Office:

2009 - 2010	Liu panshui Design Quality Review Station, Staff,
2010 – 2015	Liu panshui Planning and Design Institute, Staff.
2015 – 2022	Guizhou Institute of Technology Teachers Office, Staff/Deputy Section Chief/Section Chief.
2022 - the present	Guizhou Institute of Technology Library, Assistant Librarian