## 工贤义

## 使用工具

使用cursor进行AI编程助力, 软件主界面如下

```
[i 文件(E) 编辑(E) 选择(S) <u>查看(V)</u>
                                                                                                            ♪ OCSIP [管理员]
                                                                                                                                                                                     home > views.py > django_creat_datase
 ∨ home
                                           global dataset_id
global token
  > _pycache_
                                             if request.method == "POST":
                                             dataset_id = request.POST.get('dataset_id')
dataset_name = get_datasets(token)[dataset_id]['name']
 _init_.py
 admin.py
                                                  return render(request, 'home.html', {'dataset': get_dataset(token, dataset_id)})
 models.pv
                                      def django_login(request):
    global token
    if request.method == "POST":
 views.py
                                                 login_form = LoginForm(request.POST)
if login_form.is_valid():
 init_.py
                                                       cd = login_form.cleaned_data
                                               cd = login_form.cleaned_data
token = py_login(cd['username'], cd['password'])
if token == '':
    return HttpResponse("用户名或密码错误")
else:
    return redirect('usr', token=token)
 settings.pv
 urls.py

✓ static

                                           return HttpResponse("输入不合法")
else:
    return render(request, 'login.html')
  > data
  > EasyDL定制化模型 fi...
 大纲
                                       # 参考 (django) 01 django实现前端。
def django_upload_data(request):
                                                                                                                                      行21. 列 35 空格: 4 UTF-8 CRLF () Python 3.10.4 64-bit A OPr
```

## 使用场景

在使用python操作sqlite时,由于并未学习任何数据库的相关知识,所以本次项目中关于数据库的增删查改操作全部由cursor生成。

子场景1: 使用数据库进行account表的查找

```
    文件(E) 编辑(E) 选择(S) 查看(V) …

                                                                                                                                        © P views.py M ♣ _init_.py M ×
                                                                                                                                                        ⊳ ~ ১৯ ⊟ …
                     > migrations
                                return token if success if failed, return empty string
 admin.py
 apps.py
                                 连接数据库db.sqlite3.打开account表,读取表中的username和password_md5的所有信息,转化成字典添加到data中
Cutte Accept Cutte公 Reject Follow-up instructions... Ctrl+K
  tests.py
                                 conn = sqlite3.connect('db.sqlite3')
c = conn.cursor()
                                 c = conn.cursor()
c.execute(f"SELECT username, password_md5 FROM account")
 > _pycache_
_init_.py
                                 data = dict(c.fetchall())
conn.close()
if account in data:
                                      if data[account] == md5(pas
    token = uuid4().hex
(parameter) account: str
                                         token2account[token] = account
                                          return token

✓ static

                                     conn.commit()
                                                                                                    行 158, 列 1 空格: 4 UTF-8 CRLF ( Python 3.10.4 64-bit 🔠 🗸 Pre
```

子场景2: 使用数据库进行datasets表的增加

```
    文件(E) 编辑(E) 选择(S) 查看(V) ···

○ OCSIP [管理员]

                                                                                                                                                                                                            ⊳ ৺ ৸ 🖽 …
 OCSIP
  models.py
                                      # conn.close()
@_error("")
def creat_dataset(token: str, dataset_name: str) -> str:
    """
                                             if failed, return empty string
                                            id = uuid4().hex
                                            name = dataset_name

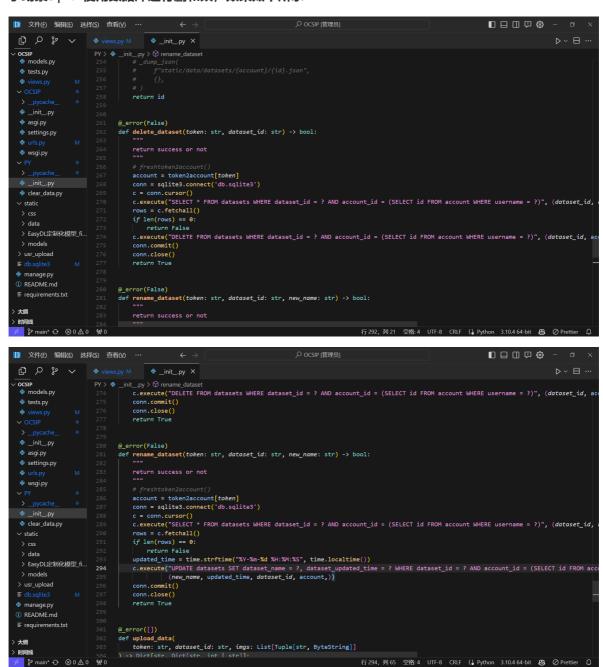
created_time = updated_time = time.strftim

"%Y-%m-%d %H:%M:%S", time.localtime())
  settings.py
                                            连接数据库db.sqlite3,打开datasets表,向表中增加一项,其中列项目分别为dataset_id,
dataset_name_dataset_created_time_dataset_updated_time_account_id,其中accunt_id的值
要任account表中进行直找,找到其username为account的项目对应的id
  clear data.py
                                             conn = sqlite3.connect('db.sqlite3')
  > data
 → EasyDL定制化模型 fi... 255
→ models 256
→ user unload 257
                                            c.execute("INSERT INTO datasets (dataset_id, dataset_name, dataset_created_time, dataset_updated_time, account_id) VALUES (?, ?, ? (id, name, created_time, updated_time, account,))
                                             conn.commit()
                              259
manage.py

≡ requirements.txt

                                            return id
〉大纲
   ழ° main* ↔ ⊗ 0 🛆 1 🕍 0
                                                                                                                                       行 259, 列 1 空格: 4 UTF-8 CRLF () Python 3.10.4 64-bit 🔠 Ø Prett
```

子场景3|4: 使用数据库进行删和改, 效果如下所示



## 使用心得

这次使用场景中使用AI助力编程能够在陌生的场景中,比如本次的数据库操作,快速的进行实现,只需要把需求,条件提供清楚,那么AI还是可以完成一份基本正确的答卷的,不然从头开始学习成本太高,也不是本次项目的中心。