**个人工作报告**

一、工作内容概述

在本次AI飞花令系统开发项目中，我作为项目组长，主要负责数据爬取、系统部署说明文档撰写，以及项目报告的统筹与核心内容编写。具体工作如下：

1. 古诗文数据爬取与存储：设计并实现爬虫程序，从古诗文网抓取唐诗数据，完成本地数据库存储；因效率问题优化数据源，切换至开源“全唐诗”数据集。

2. 系统部署文档撰写：编写详细的系统部署说明文档，涵盖环境配置、依赖安装、前后端启动流程及常见问题解决方案。

3. 项目报告统筹与撰写：协调小组成员分工，整合技术方案与实验验证结果，完成项目背景、技术路线、成果分析等核心章节的撰写。

二、工作完成过程

（一）数据爬取

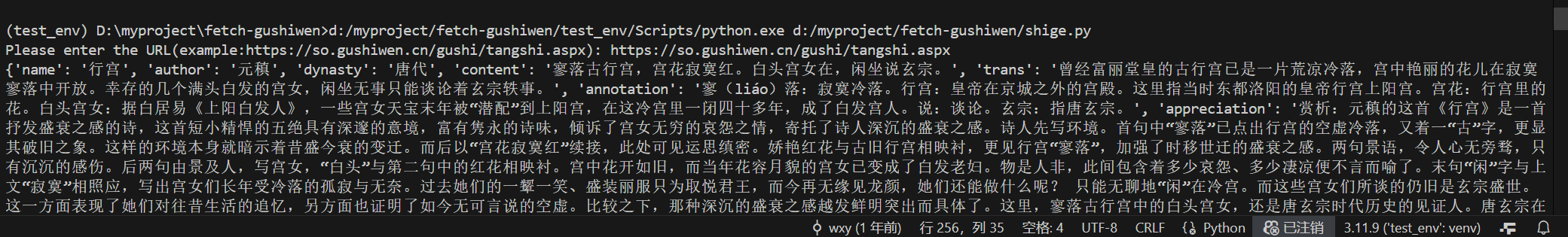
1. 爬虫程序开发

技术实现：基于Python的requests库与BeautifulSoup解析库，从古诗文网（https://so.gushiwen.cn）抓取唐诗的标题、作者、朝代、正文、注释等信息。

代码逻辑：通过fetch\_html函数获取网页内容，extract\_poem\_urls提取诗句链接，fetch\_poem\_details解析详细信息，最终通过SQLAlchemy将数据存入MySQL数据库。

爬取古诗文网的数据：

数据爬取文件：D: \nanoGPT-master\spider.py：从网页爬取古诗文信息，包括名称、作者、朝代、内容及注释等，并输出这些信息保存到本地Mysql数据库中。



原计划是通过爬虫程序，先爬取古诗文网足够多的数据后，再训练模型。但是发现这样爬取的效率太低，于是找到了开源的“全唐诗”古诗文数据集作为替代。

spider.py爬虫程序：

from datetime import datetime

from sqlalchemy.orm import sessionmaker

from model import Poem

from shige import fetch\_html, extract\_poem\_urls, fetch\_poem\_details

from utils import init\_engine

if \_\_name\_\_ == "\_\_main\_\_":

    try:

        engine = init\_engine()

        if engine is None:

            print("Failed to initialize the engine.")

            exit(1)

        url = input(

            "Please enter the URL(example:https://so.gushiwen.cn/gushi/tangshi.aspx): "

        )

        poem\_urls = []

        html\_content = fetch\_html(url)

        if html\_content:

            poem\_urls.extend(extract\_poem\_urls(html\_content))

        else:

            print("Failed to fetch or parse HTML content.")

            exit(1)

        for url in poem\_urls:

            details = fetch\_poem\_details(url)

            new\_session = sessionmaker(engine)

            with new\_session() as session:

                try:

                    poem = Poem(

                        name=details["name"],

                        author=details["author"],

                        dynasty=details["dynasty"],

                        content=details["content"],

                        trans=details["trans"],

                        annotation=details["annotation"],

                        appreciation=details["appreciation"],

                        background=details["background"],

                        created\_at=datetime.now(),

                    )

                    session.add(poem)

                    session.commit()

                    print(f"Saved details for poem: {details['name']}")

                except Exception as e:

                    session.rollback()

                    print(f"An error occurred while saving the poem: {e}")

        print("All poems saved successfully.")

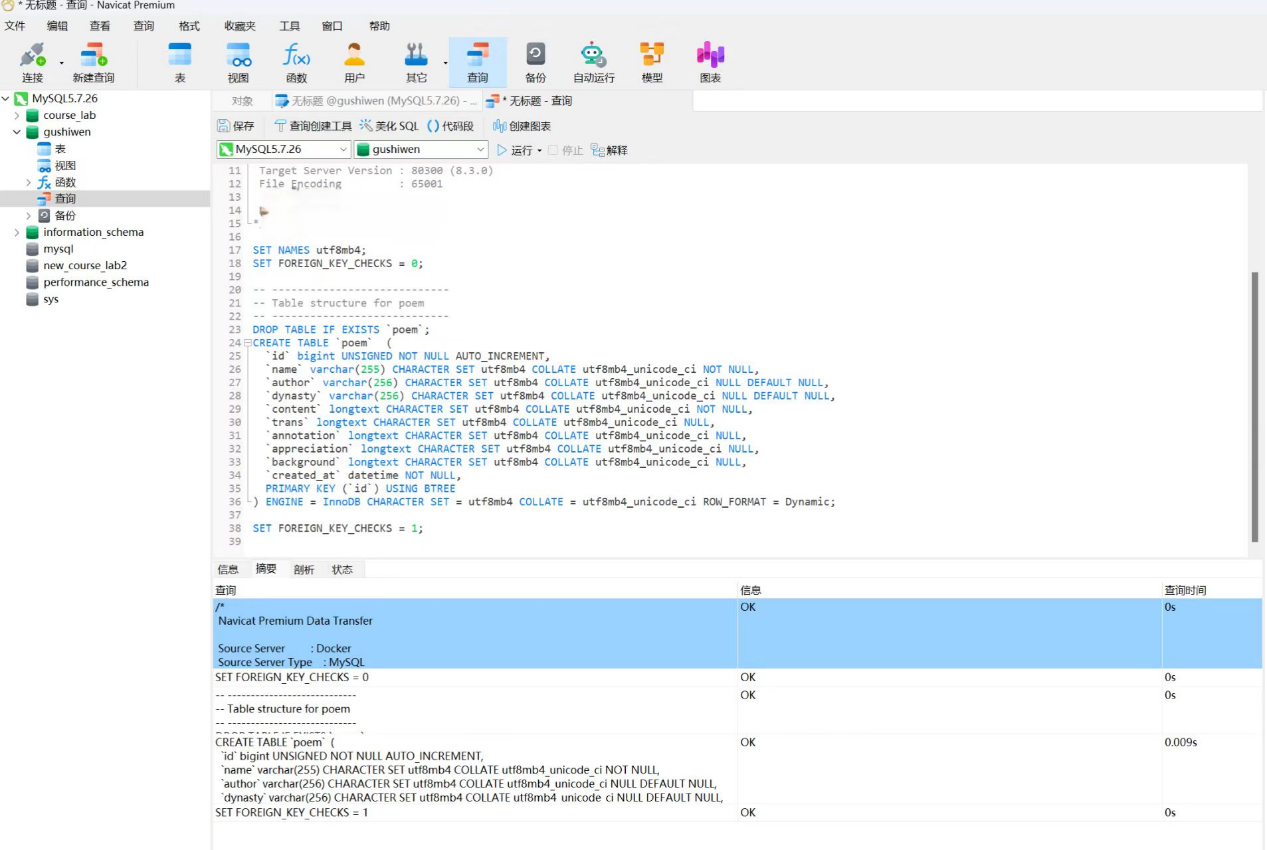
    except Exception as e:

        print(f"An error occurred: {e}")

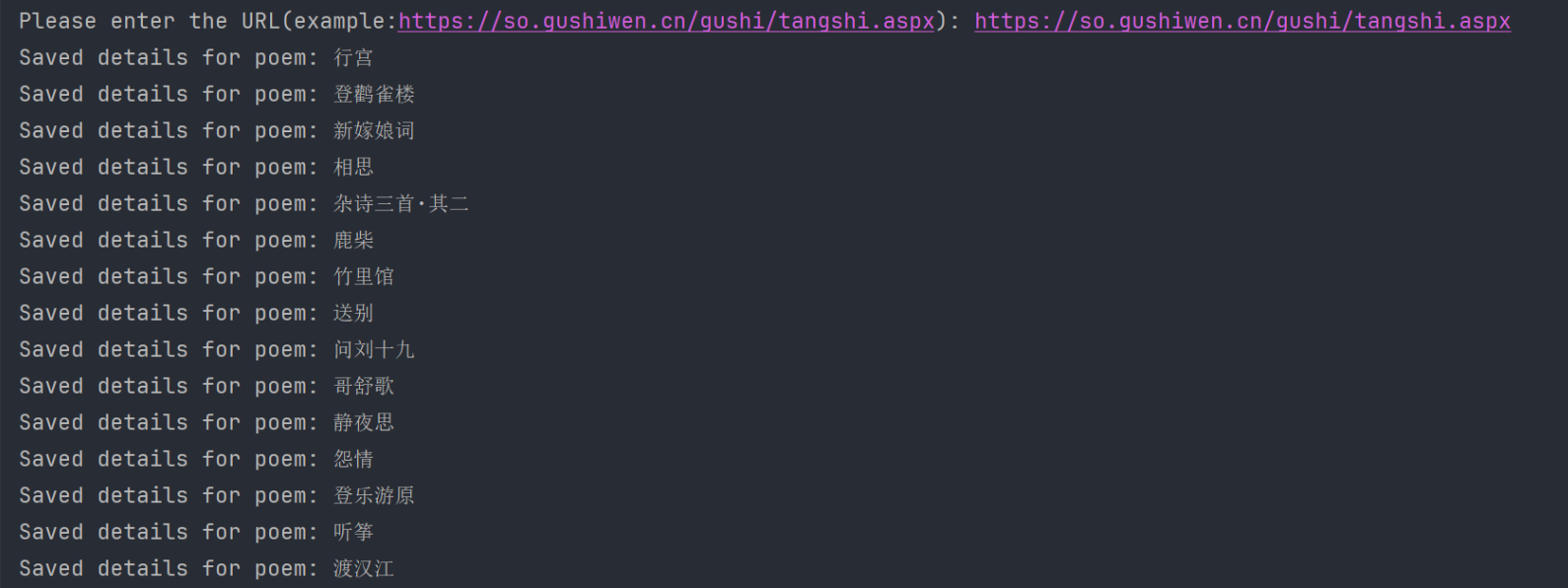
这一部分代码的目的是是从一个古诗网站抓取唐诗的诗句并将其存储到数据库中。

程序的工作流程是：初始化数据库连接，获取用户输入的 URL，抓取网页的 HTML 内容，提取诗句链接，解析每个诗句的详细信息，最后将这些信息存储到数据库中。具体步骤如下：

1.先在MySQL中建立关系型数据库。



2.然后运行spider.py爬虫程序，爬取古诗文网的数据到本地数据库中。



3.本地数据库爬取到的唐诗信息如下图所示：



挑战与优化：

初始爬虫效率低，单页解析耗时过长，无法满足数据量需求。

在小组成员黄红洲同学的帮助下，我们团队项目的数据集最终由爬取到的少量古诗文网数据，切换至开源“全唐诗”数据集，通过JSON文件直接提取五言诗，显著提升预处理效率（代码见文件`prepare\_tangshi.py`）。

（二）系统部署文档撰写

我负责爬虫程序开发，写部署说明文档，同时参与项目报告的撰写。

（三）项目报告统筹与撰写

我负责技术方案整合，提炼团队技术路线，包括AgentScope框架的智能体管理、FAISS向量检索、通义千问API调用等核心模块。

三、工作成果与收获总结

（一）工作成果

1. 数据基础构建：初期负责古诗文网的数据爬取。经过实践尝试，发现初始爬虫效率低，单页解析耗时过长，无法按照进度满足数据量需求，为后来在小组成员黄红洲的帮助下，我们团队项目的数据集切换至开源“全唐诗”数据集提供了经验根据。

2. 部署文档标准化：编写15页系统部署指南，文档涵盖环境配置、依赖安装、前后端启动命令等核心内容，降低后续维护成本。

3. 项目报告完整性：参与完成13页项目报告的撰写，将技术方案可视化呈现。

（二）收获总结

1. 技术能力提升：掌握requests库、BeautifulSoup解析、MySQL数据库交互等实战技能。学会通过开源数据集优化项目效率，平衡自研与外部资源的应用场景。

2. 项目管理经验：锻炼跨模块协作能力，通过文档标准化推动团队开发流程规范化。提升技术方案抽象与成果展示能力，为后续复杂项目奠定基础。

3. 问题解决思维：从“单一爬虫实现”到“数据源灵活切换”的优化过程，深化了对“技术为业务服务”的理解。

四、未来展望

本次项目让我意识到数据质量与工程规范对AI系统的重要性。未来计划深入学习分布式爬虫与自动化测试技术，进一步提升大规模数据处理能力，同时探索更高效的模型微调方法，以支持更复杂的交互场景。