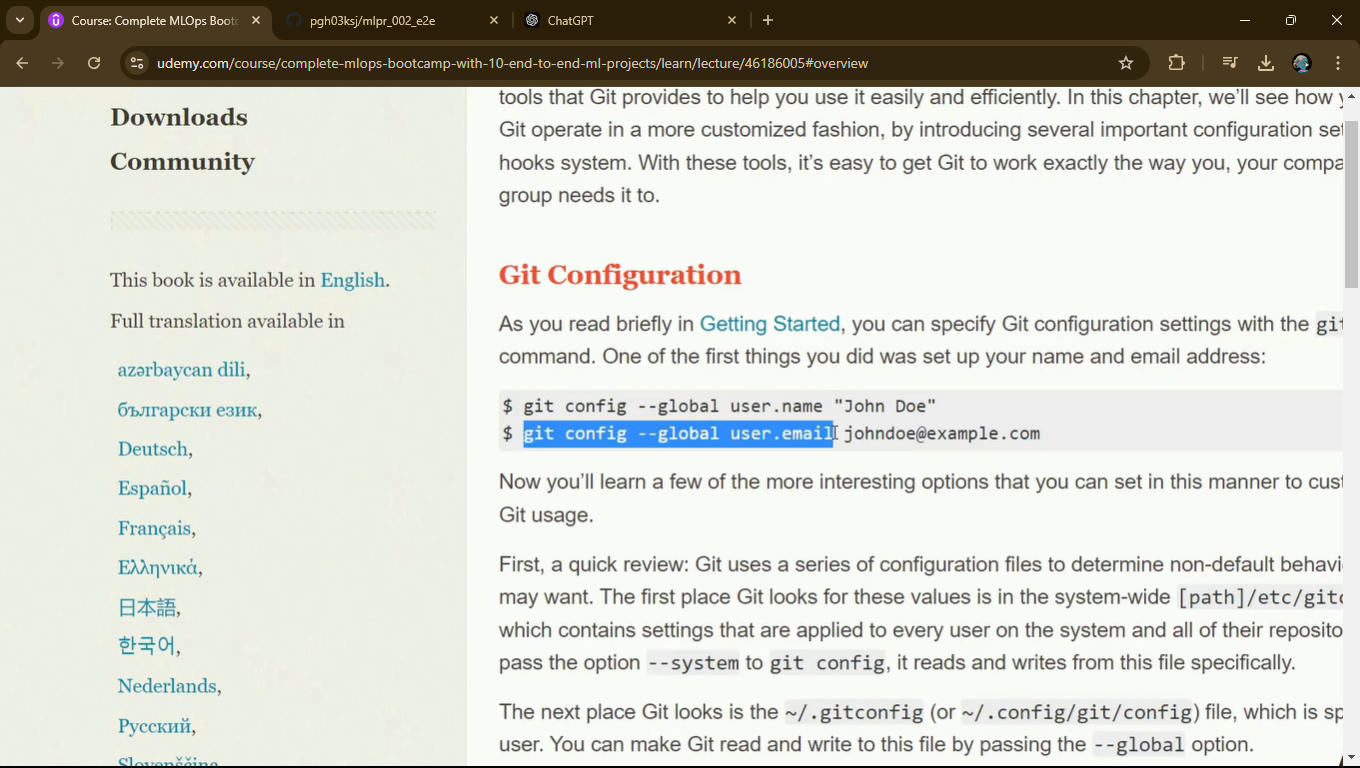
**PART 1: Setting up the Env**

1. Tạo github repo

Tạo folder làm việc  
**cd link\_to\_your\_folder  
code .**  
( Đề cử : Không làm như thế này, mà nên mở vscode rồi open folder )

1. Mở terminal (trong vsc) để sync the folder with your github repo, thực hiện các bước quen thuộc sau :  
   **\* conda create -p venv python==3.12.7 -y  
   \* conda init conda activate venv/**  
   \* Tạo file Readme  
   **\* git add README.md  
   \* git commit -m "1st commit pushing README file to repo"  
   \* git branch -M main  
   \* git remote add origin** [**https://github.com/pgh03ksj/mlpr\_002\_e2e.git**](https://github.com/pgh03ksj/mlpr_002_e2e.git) **\* git push -u origin main**  
   \* Lưu ý một bước quan trọng là đăng nhập vào email nếu chưa đăng nhập :  
     
   \* **git config --global user.email**  
   \* …
2. Tạo .gitignore ngay trên github chứ không tạo trên vsc nữa :  
   Vào repo tạo file  
   Ngay khi ghi tên file là “.gitignore”, nó sẽ hiện các option cho mình chọn.  
   \* **git pull** | Để kéo hết các file trên GH về thôi, không gì đặc biệt.  
   \*
3. Tạo setup.py & requirements.txt  
   Tạo folder “src”, và bắt buộc có file “\_\_init\_\_.py” để cái file setup.py consider đây (folder “src”) là một PACKAGE.  
   \* Xư lý 2 file trên  
   \* pip install -r requirements.txt | và see the magic !  
   ( Lưu ý rằng phải có “-e .” để file setup.py được thực thi )  
     
   \* git add .  
   \* git commit -m “second commit is setting up the setup file”  
   \* git push -u origin main

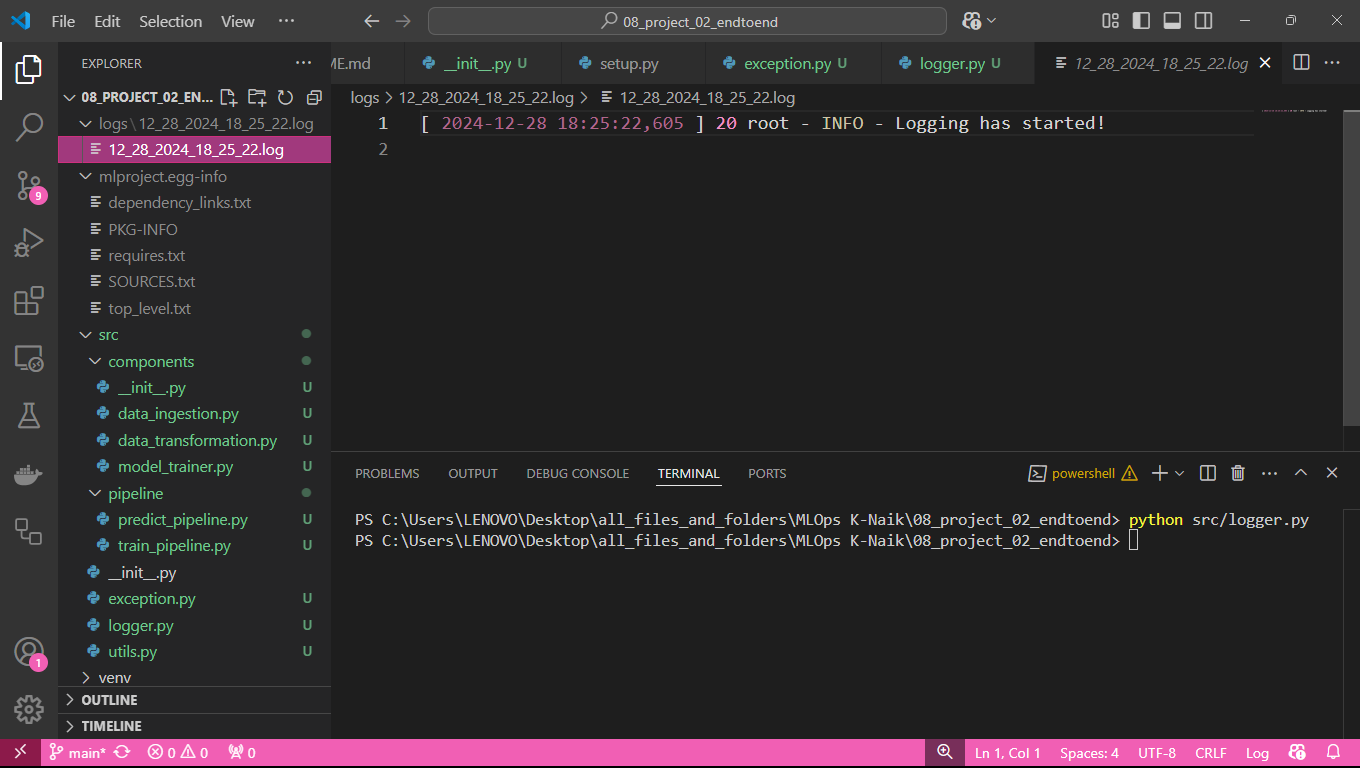
**PART 2: LOGGING & EXCEPTION**

1. Tạo các folder sau trong folder gốc “src” :  
   **src/components/**  
   src/components/\_\_init\_\_.py  
   src/components/data\_ingestion.py  
   src/components/data\_transformation.py  
   src/components/model\_trainer.py  
   **src/pipeline/**  
   src/pipeline/\_\_init\_\_.py  
   src/pipeline/train\_pipeline.py  
   src/pipeline/predict\_pipeline.py  
   **src/exception.py**  
   src/logger.py  
   src/utils.py  
   \*\* Xử lý các file **logger, exception, utils** trước. (Xem source code để biết thêm chi tiết)

Thêm hàng này vào file logger để có thể test xem mọi thứ hoạt động được không.

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

    logging.info("Logging has started!")

* \* python src/logger.py  
  Nếu có folder log được tạo ra :  
  

Thì coi như hoạt động được.  
Tương tự cho file exception.py :

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

    try:

        a=1/0

    except Exception as e:

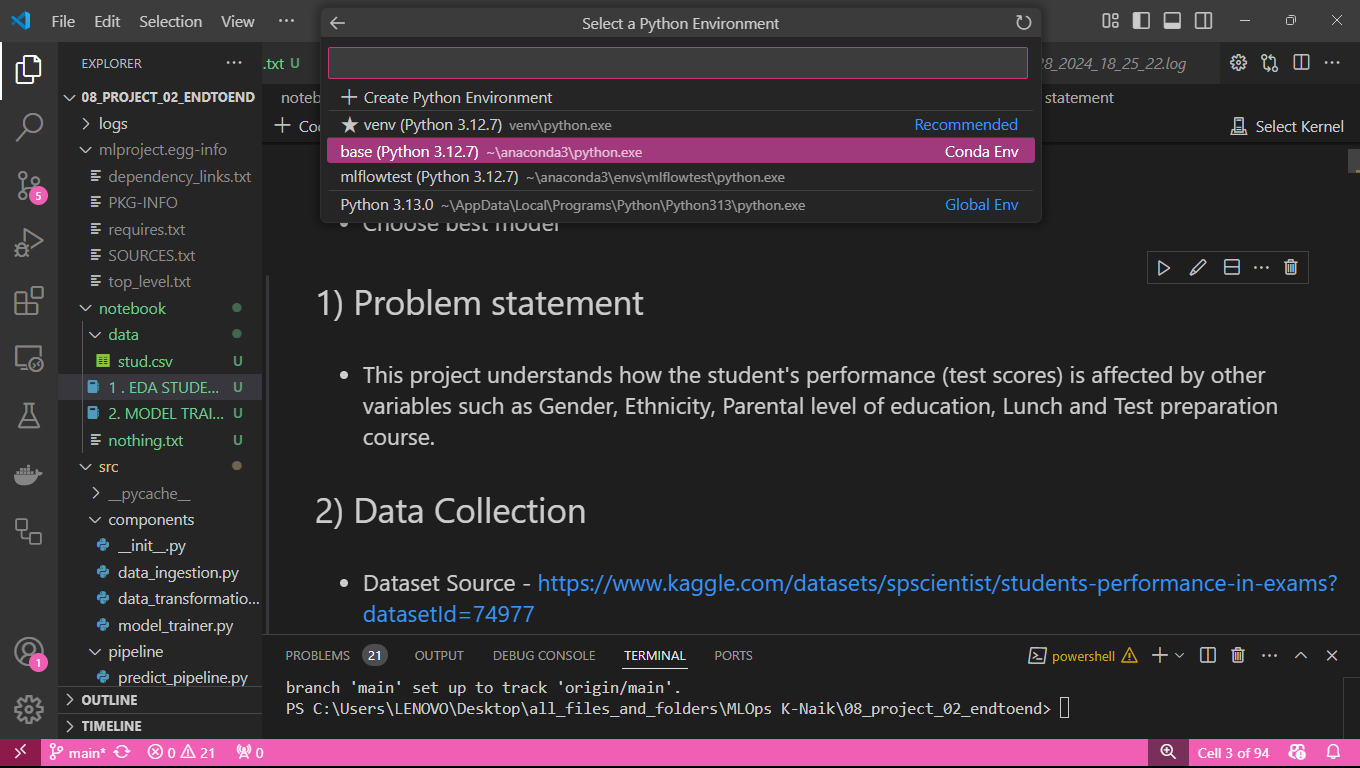
        logging.info("Division is not available !")

        raise CustomException(e, sys)

# Chạy: python src/exception.py

* Giờ thêm file vào git thôi :  
  \* git status  
  \* git add .  
  \* git commit -m “Third (3) commit create logger, exception files”  
  \* git push -u origin main

**PART 3: EDA & MODEL TRAINING**

1. Lưu ý về kernel, cái kernel để chạy được file .ipynb nó phải như thế này :  
     
     
   \* Các bước EDA hay train test gì đó là quá đơn giản với MLe.  
   \* Sau khi execute xong các cell, đưa nó lên git.   
   \* Nhắc lại một lần nữa :   
   **`git add .`   
   `git status`   
   `git commit -m "Fourth commit, done some EDA and train different models"`  
   `git push -u origin main`**

**PART 4: DATA INGESTION**

1. Xem kỹ file Data Ingestion trong source code để tìm hiểu kỹ hơn.  
   Sau khi copy dán vào rồi thì chạy thử :  
   \* python src/components/data\_ingestion.py  
   ( Nếu chưa chạy được thì phải fix, thường là fix cơ bản như sau :  
   `**pip install pandas**`  
   `**pip install scikit-learn**`  
   `**cd "C:\Users\LENOVO\Desktop\all\_files\_and\_folders\MLOps K-Naik\08\_project\_02\_endtoend**"  
   **python -m src.components.data\_ingestion**`  
     
     
     
   \* Tất cả các bước trên đều không thành công vì có một thời điểm cần execute file “data\_ingestion.py” khi code chưa được hoàn chỉnh như trong source code của Krish.  
   ***\* Đó là bước mà khi thực thi xong, ta tạo ra được artifacts/các\_csv\_files  
   \* Bước tiếp theo của bước đó, là vào “.gitignore” để add “.artifacts” vào chỗ # Enviroment***
2. Copy 2 file còn lại (data\_transformation.py, utils.py (trong folder “pipeline”) )  
   \* Chú ý install thêm “dill”

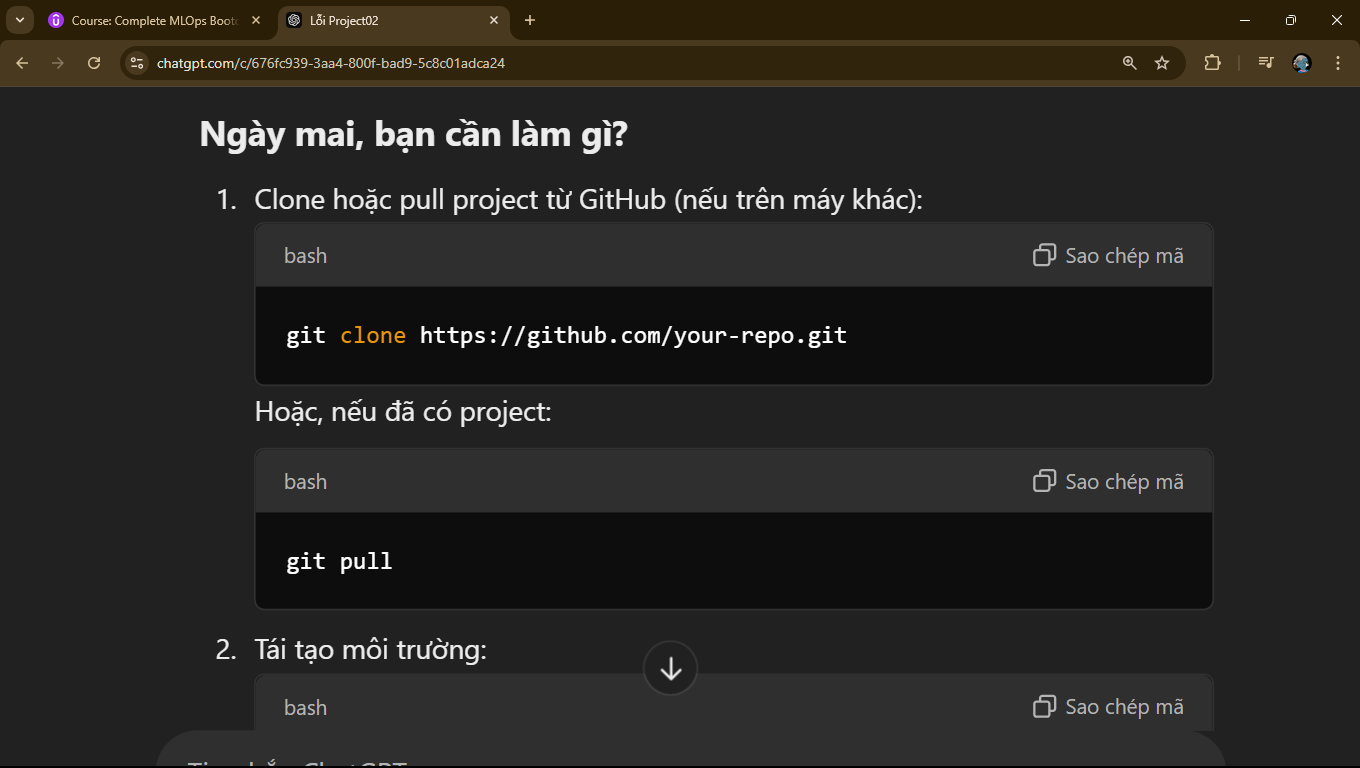
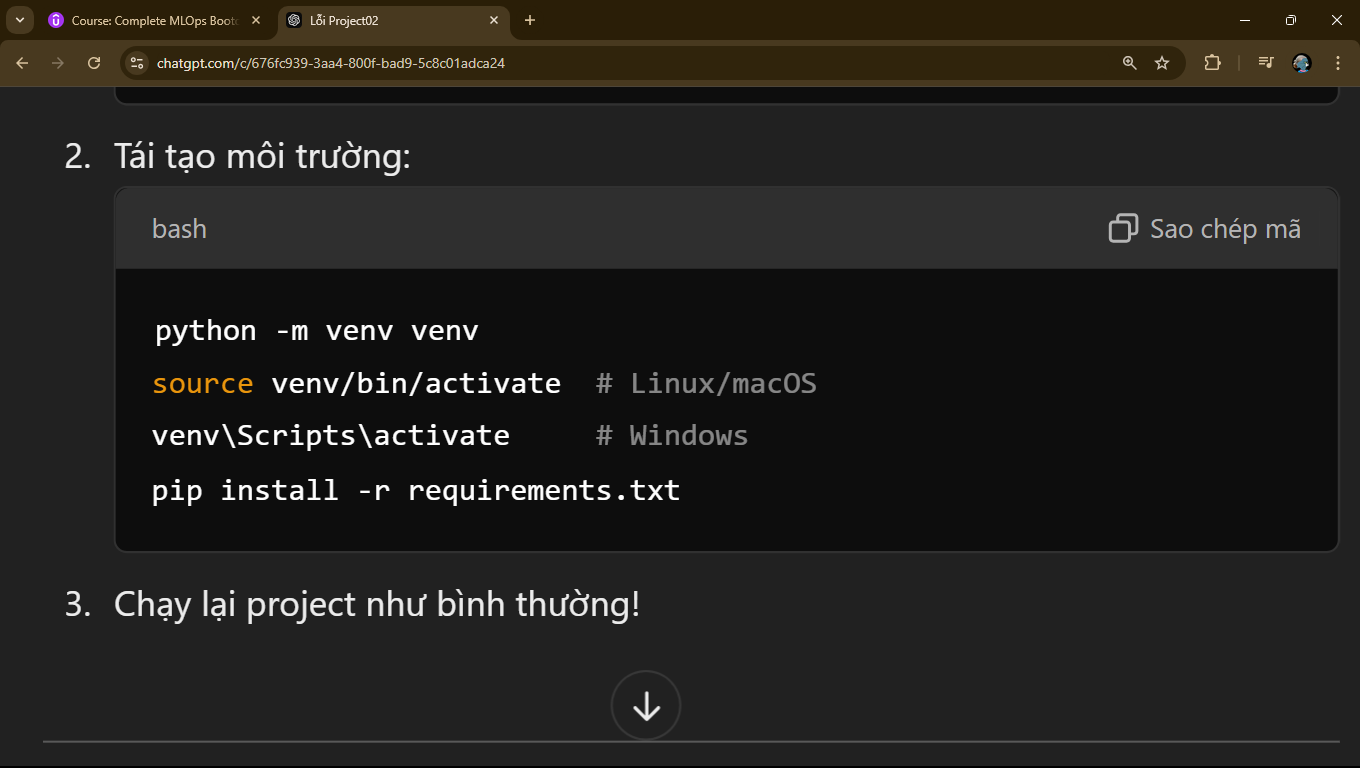
…

…

…

1. Có quá nhiều lỗi thực thi trong quá trình làm, may mắn thay, chatgpt đã giải quyết được hết. Code đã được lưu trữ lại trong folder, tham khảo thêm. Lưu ý một số lưu ý như sau:  
   (1) scikit-learn==1.6.0

**Lưu ý quan trọng :**

* Lỡ tắt VSCODE rồi thì làm các bước sau đây:
  + Bước 1:
    - 
  + Bước 2:
    - 

1. Sau khi copy cả các file như app.py và cả folder templates chứa html vào. Ta tiến hành :  
   `python app.py`  
   Truy cập :  
   (1) http://127.0.0.1:5000/  
   (2) <http://127.0.0.1:5000/predictdata>  
   \*\* Lưu ý nhớ bật chế độ `debug = True` để phát hiện lỗi hẳn hoi trên trang web để có thể dễ sửa. Nhưng khi deploy với Elastic Beanstalk, ta phải gỡ nó đi.
2. Deploy với ElasticBeanstalk :

* Bổ sung thêm folder:   
  C:\Users\LENOVO\Desktop\all\_files\_and\_folders\MLOps K-Naik\08\_project\_02\_endtoend\**.ebextensions\python.config**
* `**git add .**`
* Lỗi ngoài ý muốn :
  + \*\* **rm -Force .git/index.lock** || dùng để giải quyết lỗi xảy ra khi một quá trình Git khác đang chạy, hoặc một quá trình trước đó đã gặp sự cố và không đóng đúng cách, khiến file index.lock không bị xóa.
* `**git commit -m "sixth commit, deployment configuration"**`
* `**git push -u origin main**`

**🡪 Giờ đến phần sử dụng acc AWS :**

* + Bước 1: Tìm kiếm Elastic Beanstalk trong AWS Dashboard.
  + Bước 2: Create Application
  + Bước 3: Cài đặt các cài đặt sau :  
    (\*) Application Name : studentperformance  
    (\*) Platform : Python  
    (\*) Application Code : Sample Application