Nguyễn Đức Toàn - 22110437

from flask import Flask, render\_template, request

import subprocess

app = Flask(\_\_name\_\_)

# Các file JSON có thể chọn

json\_files = [

    "input/books\_clean.json",

    "input/tiki\_book\_data\_clean.json"

]

# Danh sách các chức năng MapReduce

jobs = {

    "avg\_rating": {

        "mapper": "mapper/avg\_rating\_mapper.py",

        "reducer": "reducer/avg\_rating\_reducer.py",

        "input\_type": "single"

    },

    "max\_price": {

        "mapper": "mapper/max\_price\_mapper.py",

        "reducer": "reducer/max\_price\_reducer.py",

        "input\_type": "single"

    },

    "min\_price": {

        "mapper": "mapper/min\_price\_mapper.py",

        "reducer": "reducer/min\_price\_reducer.py",

        "input\_type": "single"

    },

    "count\_rating": {

        "mapper": "mapper/count\_rating\_mapper.py",

        "reducer": "reducer/count\_rating\_reducer.py",

        "input\_type": "single"

    },

    "avg\_price": {

        "mapper": "mapper/avg\_price\_mapper.py",

        "reducer": "reducer/avg\_price\_reducer.py",

        "input\_type": "single"

    },

    "compare\_avg\_price": {

        "mapper": "mapper/compare\_avg\_price\_mapper.py",

        "reducer": "reducer/compare\_avg\_price\_reducer.py",

        "input\_type": "multiple"

    }

}

def run\_mapreduce(mapper, reducer, input\_files, job\_name=None):

    try:

        if job\_name == "compare\_avg\_price":

            mapper\_outputs = []

            tags = ["books", "tiki"]

            for i, file in enumerate(input\_files):

                with open(file, "r", encoding="utf-8") as f:

                    file\_content = f.read()

                mapper\_proc = subprocess.Popen(

                    ["python3", mapper, tags[i]],

                    stdin=subprocess.PIPE,

                    stdout=subprocess.PIPE,

                    stderr=subprocess.PIPE,

                    text=True

                )

                mapper\_output, mapper\_err = mapper\_proc.communicate(input=file\_content)

                if mapper\_proc.returncode != 0:

                    return f"Mapper error for {tags[i]}:\n{mapper\_err}"

                mapper\_outputs.append(mapper\_output)

            combined\_output = "\n".join(mapper\_outputs)

            reducer\_proc = subprocess.Popen(

                ["python3", reducer],

                stdin=subprocess.PIPE,

                stdout=subprocess.PIPE,

                stderr=subprocess.PIPE,

                text=True

            )

            reducer\_output, reducer\_err = reducer\_proc.communicate(input=combined\_output)

            if reducer\_proc.returncode != 0:

                return f"Reducer error:\n{reducer\_err}"

            return reducer\_output

        # Xử lý các job dạng single

        if isinstance(input\_files, list):

            combined\_input = ""

            for file in input\_files:

                with open(file, "r", encoding="utf-8") as f:

                    combined\_input += f.read()

        else:

            with open(input\_files, "r", encoding="utf-8") as f:

                combined\_input = f.read()

**# Chạy mapper**

        mapper\_proc = subprocess.Popen(

            ["python3", mapper],

            stdin=subprocess.PIPE,

            stdout=subprocess.PIPE,

            stderr=subprocess.PIPE,

            text=True

        )

        mapper\_output, mapper\_err = mapper\_proc.communicate(input=combined\_input)

        if mapper\_proc.returncode != 0:

            return f"Mapper error:\n{mapper\_err}"

        # Với job như count\_rating, cần sort output

        if job\_name == "count\_rating":

            mapper\_output = '\n'.join(sorted(mapper\_output.strip().splitlines()))

**# Chạy reducer**

        reducer\_proc = subprocess.Popen(

            ["python3", reducer],

            stdin=subprocess.PIPE,

            stdout=subprocess.PIPE,

            stderr=subprocess.PIPE,

            text=True

        )

        reducer\_output, reducer\_err = reducer\_proc.communicate(input=mapper\_output)

        if reducer\_proc.returncode != 0:

            return f"Reducer error:\n{reducer\_err}"

        return reducer\_output

    except Exception as e:

        return f"Exception occurred:\n{str(e)}"

@app.route("/", methods=["GET", "POST"])

def index():

    result = None

    selected\_job = None

    selected\_file = None

    disable\_file\_select = False

    if request.method == "POST":

        selected\_job = request.form.get("job")

        selected\_file = request.form.get("input\_file")

        if selected\_job in jobs:

            job = jobs[selected\_job]

            if job["input\_type"] == "single":

                if selected\_file not in json\_files:

                    result = "Vui lòng chọn file hợp lệ."

                else:

                    result = run\_mapreduce(job["mapper"], job["reducer"], selected\_file, job\_name=selected\_job)

            else:

                # Nếu multiple thì tự động chọn cả 2 file

                result = run\_mapreduce(job["mapper"], job["reducer"], json\_files, job\_name=selected\_job)

            disable\_file\_select = (job["input\_type"] == "multiple")

        else:

            result = "Chức năng không hợp lệ."

    return render\_template(

        "index.html",

        jobs=jobs.keys(),

        files=json\_files,

        result=result,

        selected\_job=selected\_job,

        selected\_file=selected\_file,

        disable\_file\_select=disable\_file\_select

    )

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

    app.run(debug=True)