**Xây dựng các chương trình mapreduce**

**Những công việc có lương hơn 10 triệu**

Code của map.py:

#!/usr/bin/env python

import sys

import csv

# Đọc dữ liệu đầu vào

reader = csv.reader(sys.stdin)

# Bỏ qua dòng header

header = next(reader)

for line in reader:

    try:

        company\_name = line[0].strip()  # Tên công ty ở vị trí index 0

        title = line[1].strip()  # Tiêu đề công việc ở vị trí index 1

        salary\_raw = line[3].strip()  # Lương raw ở vị trí index 3

        salary\_normalized = line[4].strip()  # Lương đã chuẩn hóa ở vị trí index 4

        salary\_unit = line[5].strip()  # Đơn vị tiền tệ ở vị trí index 5

        # Kiểm tra nếu lương có giá trị và là dạng số

        if salary\_normalized.isdigit():

            salary = int(salary\_normalized)

            if 10000000 <= salary :

                # Chỉ xuất tên công ty, tiêu đề công việc, và lương

                print(f"{title}\t{salary\_normalized}")

    except Exception as e:

        # Nếu có lỗi thì in thông báo lỗi

        print(f"Lỗi khi xử lý dòng: {line}. Lỗi: {e}")

Code reduce.py

#!/usr/bin/env python

import sys

# Đọc đầu vào từ stdin (output từ map.py)

for line in sys.stdin:

    # Output kết quả ra màn hình

    print(line.strip())

**Những Công ty ở Hà Nội hoặc Hồ Chí Minh**

Code của map.py:

#!/usr/bin/env python3

import sys

import csv

reader = csv.reader(sys.stdin)

next(reader)  # bỏ dòng header

for line in reader:

    try:

        company\_name = line[1].strip()

        location = line[6].strip().lower()  # chuyển về lower để so sánh dễ

        if "hà nội" in location or "ho chi minh" in location or "hồ chí minh" in location:

            print(f"{company\_name}\t{location}")

    except Exception as e:

        continue

Code reduce.py

#!/usr/bin/env python

import sys

# Đọc đầu vào từ stdin (output từ map.py)

for line in sys.stdin:

    # Output kết quả ra màn hình

    print(line.strip())

**Những công việc không yêu cầu kinh nghiệm**

Code của map.py:

import sys

import csv

import re

reader = csv.reader(sys.stdin)

header = next(reader)

for line in reader:

    try:

        title = line[1].strip()

        experience = line[7].strip().lower()

        # Nếu không yêu cầu kinh nghiệm

        if "không yêu cầu" in experience:

            print(f"{title}\t{experience}")  # In với tab

            continue

        # Dùng regex tìm số trong chuỗi

        match = re.search(r"\d+", experience)

        if match:

            year\_str = match.group(0)

            if year\_str in ["0"]:

                print(f"{title}\t{experience}")  # In với tab

    except Exception as e:

        print(f"Lỗi khi xử lý dòng: {line}. Lỗi: {e}")

Code reduce.py

#!/usr/bin/env python

import sys

# Đọc đầu vào từ stdin (output từ map.py)

for line in sys.stdin:

    # Output kết quả ra màn hình

    print(line.strip())

**Những công việc cần ngôn ngữ python**

Code của map.py:

import sys

import csv

reader = csv.reader(sys.stdin)

header = next(reader)  # Bỏ qua dòng tiêu đề

for line in reader:

    try:

        if len(line) < 15:

            continue  # Bỏ qua nếu dòng không đủ cột

        title = line[1].strip()  # Tên công việc

        skills\_raw = line[14].strip().lower()  # Cột kỹ năng

        # Nếu cột kỹ năng rỗng thì bỏ qua

        if not skills\_raw:

            continue

        # Tách kỹ năng và loại bỏ khoảng trắng dư

        skills = [skill.strip() for skill in skills\_raw.split(",")]

        # Kiểm tra xem có chứa 'python' hay không

        if any("python" in skill for skill in skills):

            print(f"{title}\t{', '.join(skills)}")

    except Exception as e:

        print(f"Lỗi khi xử lý dòng: {line}. Lỗi: {e}", file=sys.stderr)

Code reduce.py

#!/usr/bin/env python

import sys

# Đọc đầu vào từ stdin (output từ map.py)

for line in sys.stdin:

    # Output kết quả ra màn hình

    print(line.strip())

**Lương trung bình theo từng thành phố**

Code map.py

import sys import csv

reader = csv.reader(sys.stdin) header = next(reader)

for line in reader: try: if len(line) < 7: continue

   city = line[6].strip()   
    salary = line[4].strip()   
   
    # Bỏ qua nếu chứa nhiều tỉnh/thành phố hoặc mô tả mơ hồ   
    if any(x in city.lower() for x in ['&', ',', 'nơi khác']):   
        continue   
   
    if not salary:   
        continue   
   
    salary = float(salary)   
    print(f"{city}\t{salary}")   
except Exception as e:   
    print(f"Lỗi dòng: {line}. Lỗi: {e}", file=sys.stderr) 

Code reduce.py

import sys

current\_city = None salary\_sum = 0.0 count = 0

for line in sys.stdin: try: city, salary = line.strip().split('\t') salary = float(salary)

   if current\_city == city:   
        salary\_sum += salary   
        count += 1   
    else:   
        if current\_city:   
            avg\_salary = salary\_sum / count   
            print(f"{current\_city}\t{avg\_salary:.2f}")   
        current\_city = city   
        salary\_sum = salary   
        count = 1   
except Exception as e:   
    print(f"Lỗi dòng: {line}. Lỗi: {e}", file=sys.stderr) 

if current\_city: avg\_salary = salary\_sum / count print(f"{current\_city}\t{avg\_salary:.2f}")

**Top 10 công việc có mức lương cao nhất**

Code map.py

import sys import csv

reader = csv.reader(sys.stdin) header = next(reader)

for line in reader: try: if len(line) < 5: continue

   title = line[1].strip()   
    salary = line[4].strip()   
   
    if not salary:   
        continue   
   
    salary = float(salary)   
    print(f"{title}\t{salary}")   
except Exception as e:   
    print(f"Lỗi dòng: {line}. Lỗi: {e}", file=sys.stderr)

Code reduce.py

import sys import heapq

top\_jobs = []

for line in sys.stdin: try: title, salary = line.strip().rsplit('\t', 1) salary = float(salary)

   # Thêm vào heap   
    heapq.heappush(top\_jobs, (salary, title))   
   
    # Nếu > 10 phần tử thì bỏ phần tử lương nhỏ nhất   
    if len(top\_jobs) > 10:   
        heapq.heappop(top\_jobs)   
except Exception as e:   
    continue  # Bỏ qua lỗi để không dừng chương trình 

for salary, title in sorted(top\_jobs, reverse=True): print(f"{title}\t{int(salary)}")

**Những kĩ năng tuyển dụng theo số lượng**

Code map.py

import sys

# Để bỏ qua dòng đầu tiên

first\_line = True

# Đọc input từ STDIN (Standard Input)

for line in sys.stdin:

    line = line.strip()

    # Bỏ qua dòng đầu tiên (header)

    if first\_line:

        first\_line = False

        continue

    # Chia cột dựa trên dấu phẩy (CSV format), bạn có thể thay đổi nếu dữ liệu có định dạng khác

    fields = line.split(',')

    if len(fields) > 1:

        group = fields[0]  # Giả sử nhóm công việc là trường đầu tiên

        # Emit group và giá trị là 1 (để đếm)

        print(f"{group}\t1")

Code reduce.py

import sys

current\_group = None current\_count = 0

for line in sys.stdin: line = line.strip()

# Tách group và count từ các cặp key-value (group, count)   
group, count = line.split('\t')   
   
# Chuyển count thành số   
try:   
    count = int(count)   
except ValueError:   
    continue   
   
# Nếu group thay đổi, in kết quả   
if current\_group == group:   
    current\_count += count   
else:   
    if current\_group:   
        print(f"{current\_group}\t{current\_count}")   
    current\_group = group   
    current\_count = count 

if current\_group == group: print(f"{current\_group}\t{current\_count}")

**Top 10 thành phố có nhiều việc làm nhất**

code map.py

#!/usr/bin/env python3

import sys

import csv

import re

invalid\_keywords = ['khác', 'nơi khác']

reader = csv.reader(sys.stdin)

for fields in reader:

    if len(fields) < 6:

        continue

    dia\_diem = fields[6]  # Thay bằng cột chứa "thành phố" trong dữ liệu của bạn

    parts = re.split(r'[,&]| và ', dia\_diem)

    for part in parts:

        part = part.strip()

        if any(invalid in part.lower() for invalid in invalid\_keywords):

            continue

        if part:

            print(f"{part}\t1")

Code reduce .py

#!/usr/bin/env python3 import sys from collections import defaultdict

city\_counts = defaultdict(int)

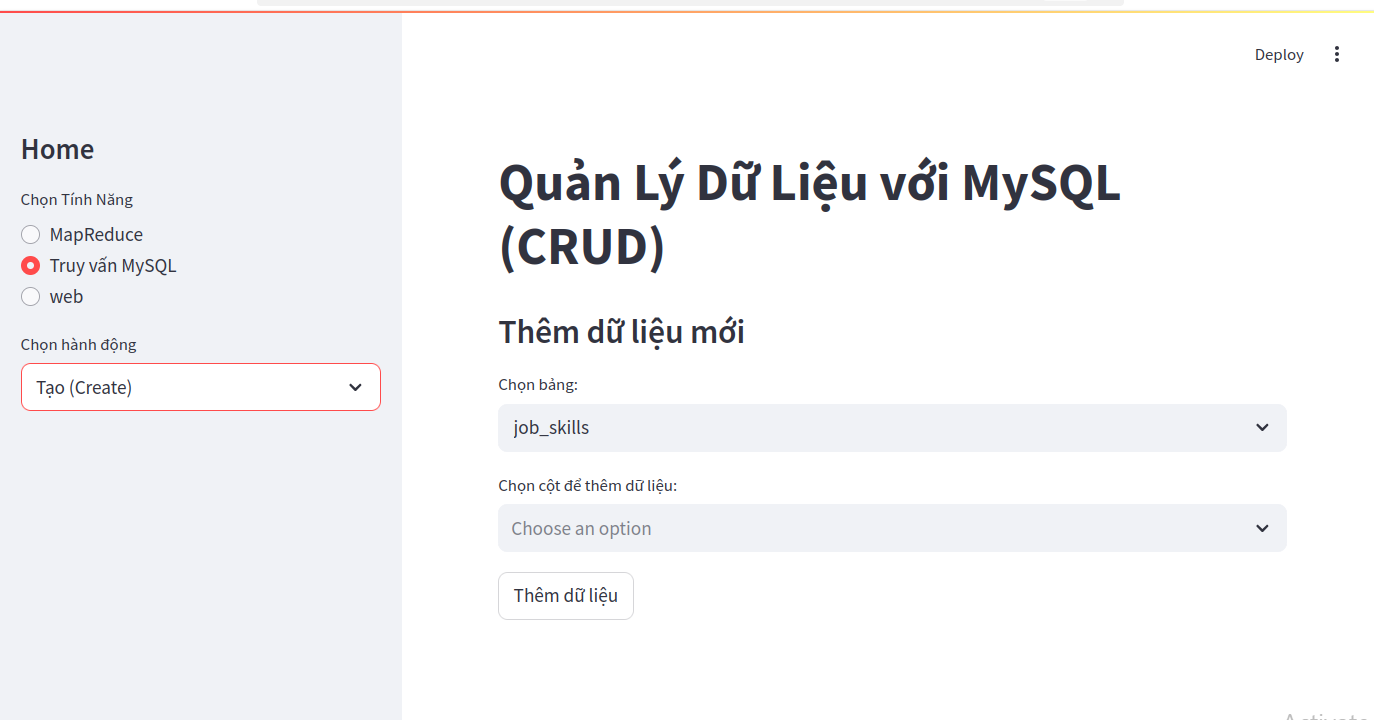
for line in sys.stdin: line = line.strip() if not line: continue city, count = line.split("\t") city\_counts[city] += int(count)

top\_10 = sorted(city\_counts.items(), key=lambda x: x[1], reverse=True)[:10]

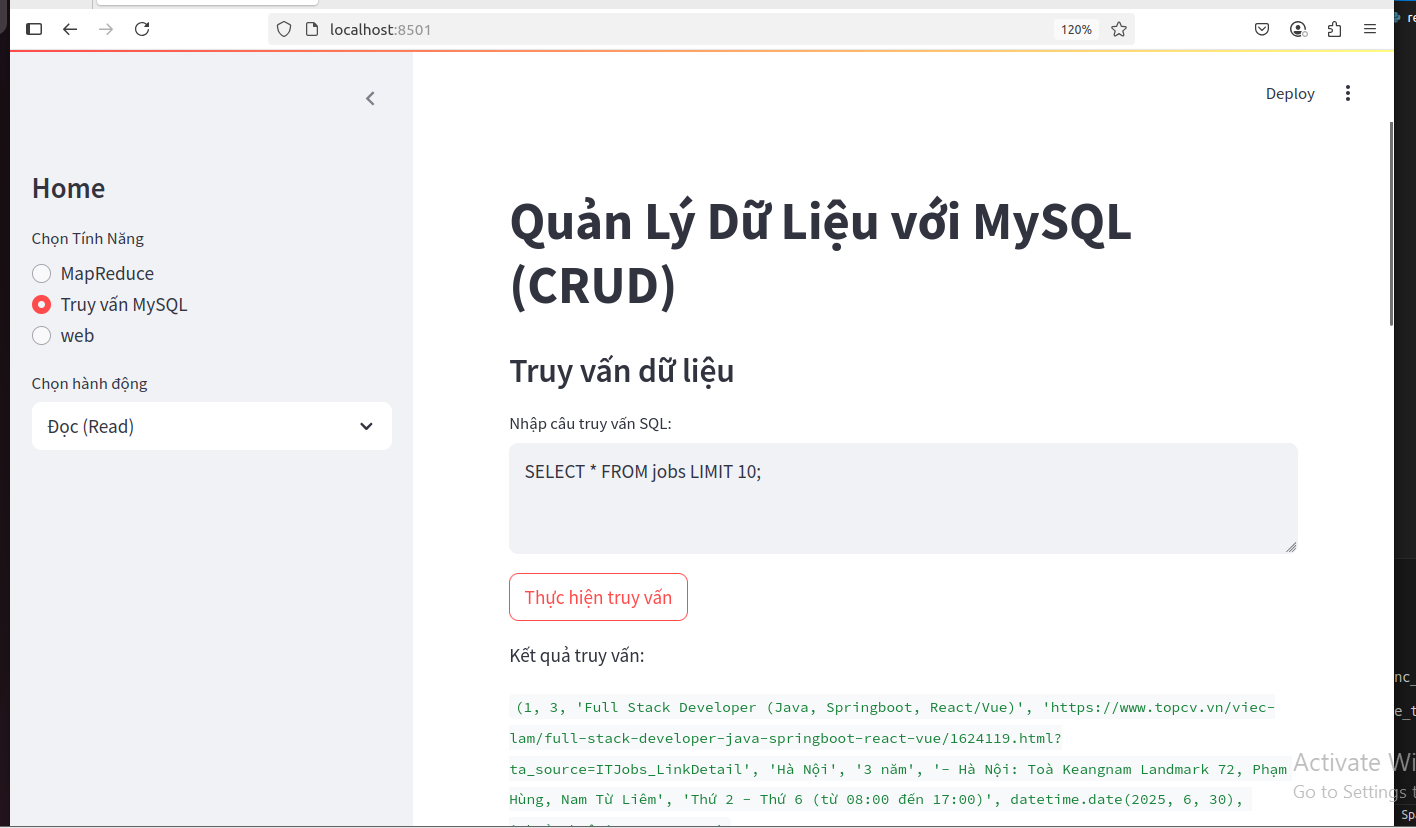
for city, total\_count in top\_10: print(f"{city}\t{total\_count}")

**Hỗ trợ truy vấn dữ liệu CRUD trên giao diện**

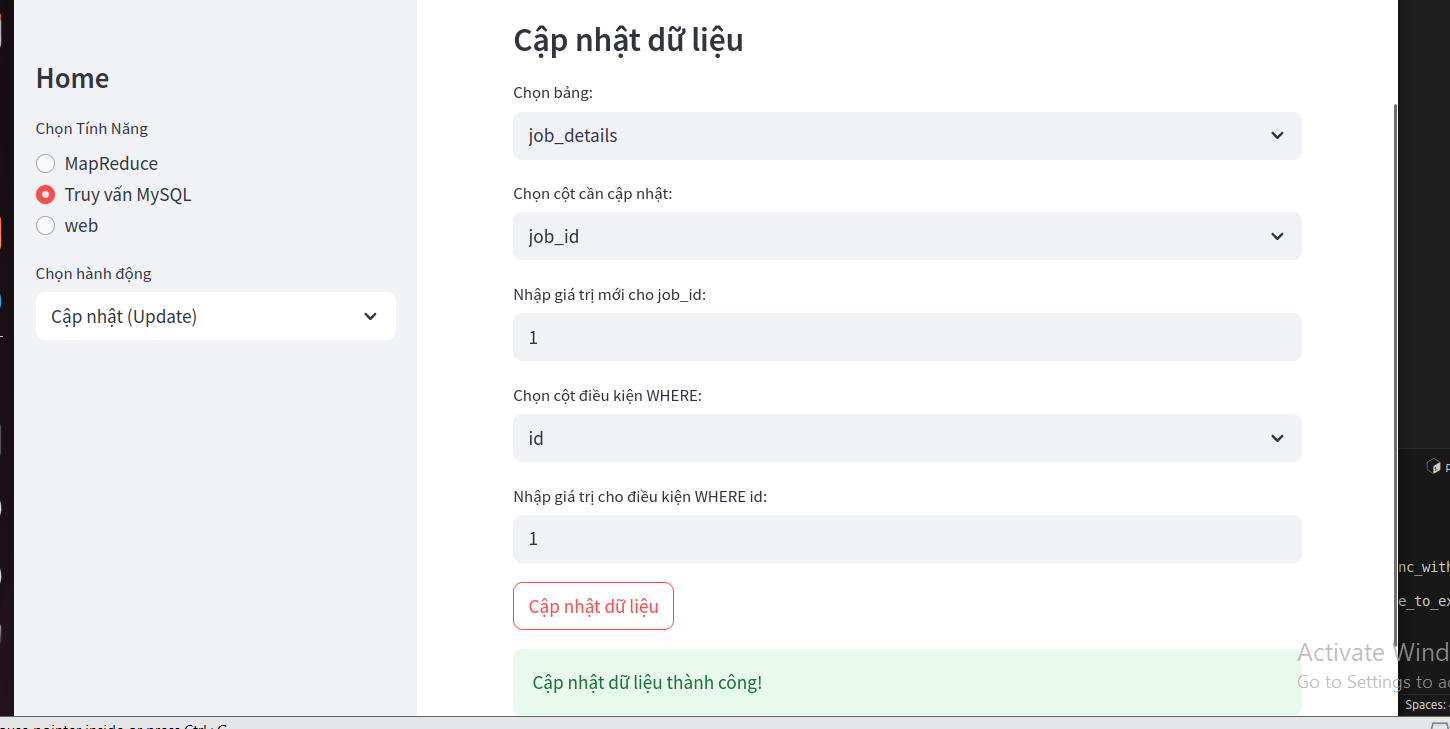
**Thêm Dữ liệu:**



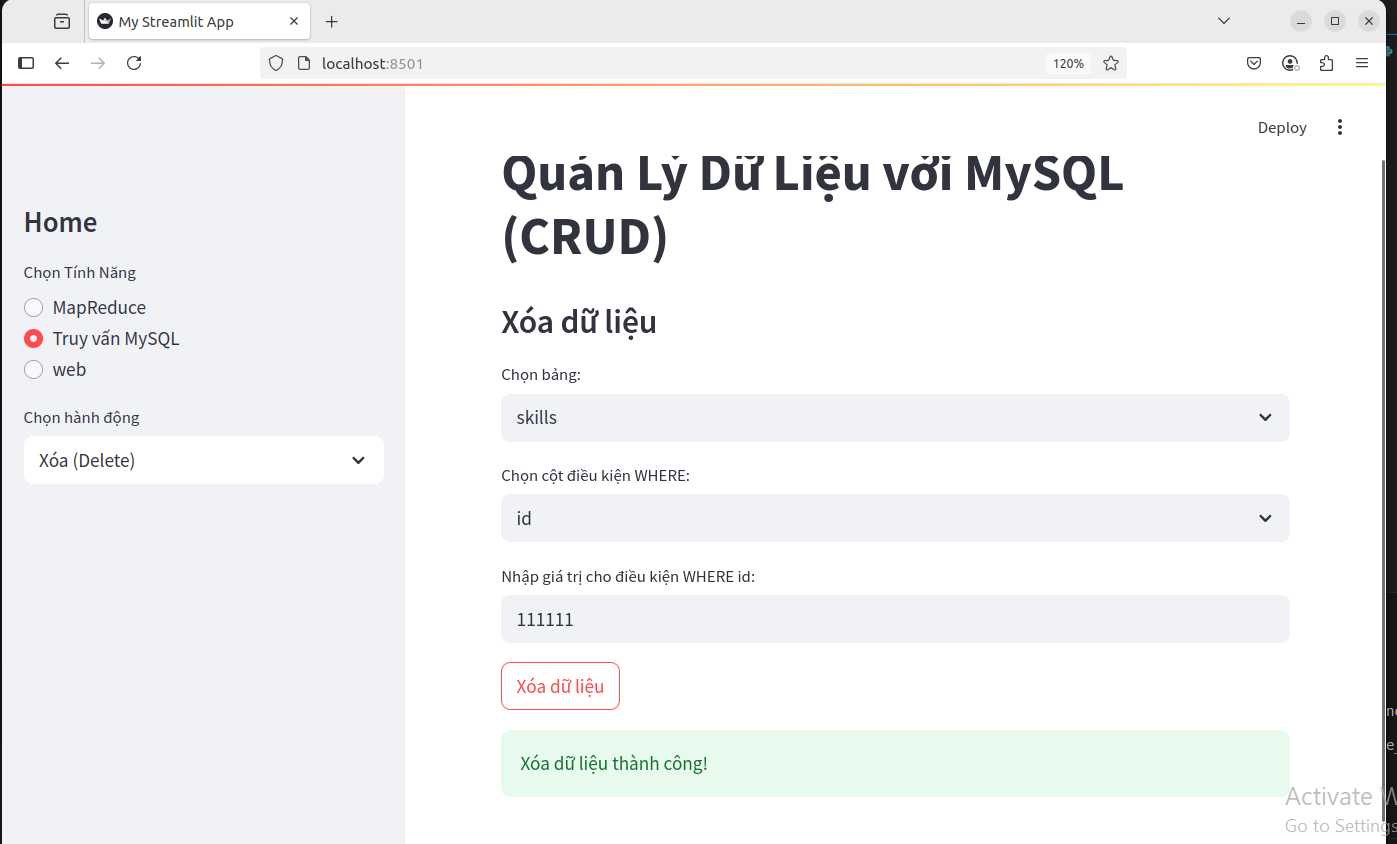
**Đọc Dữ liệu:**



**Cập nhật dữ liệu:**

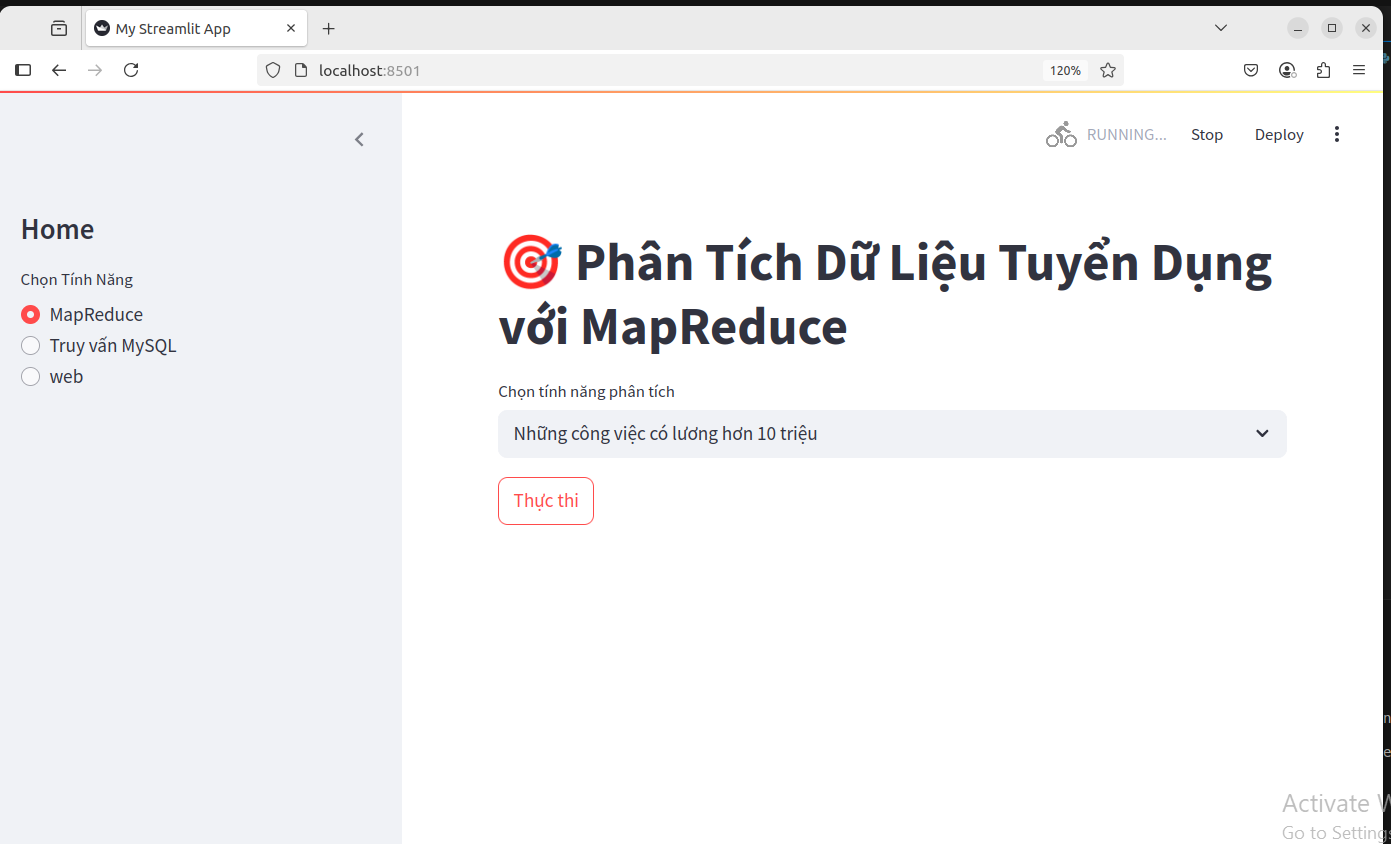


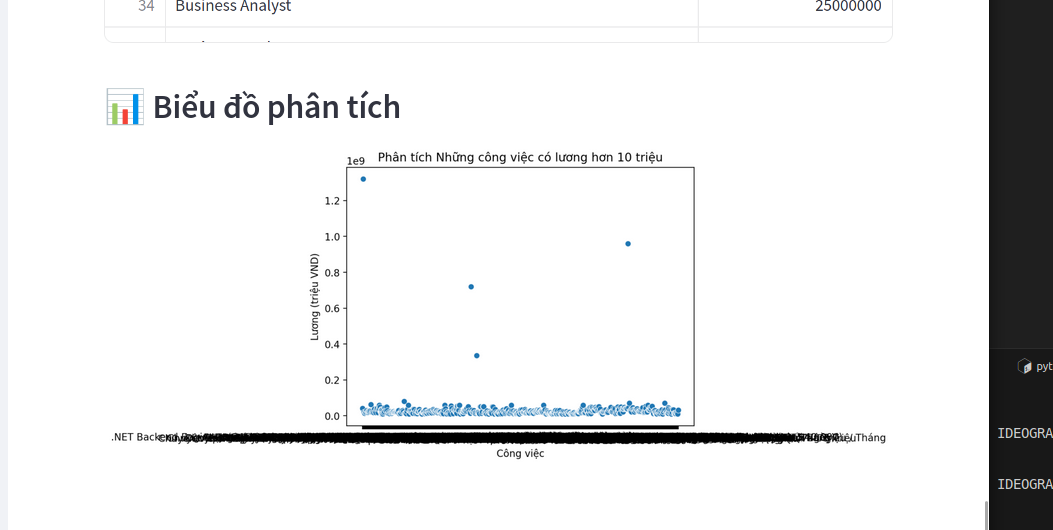
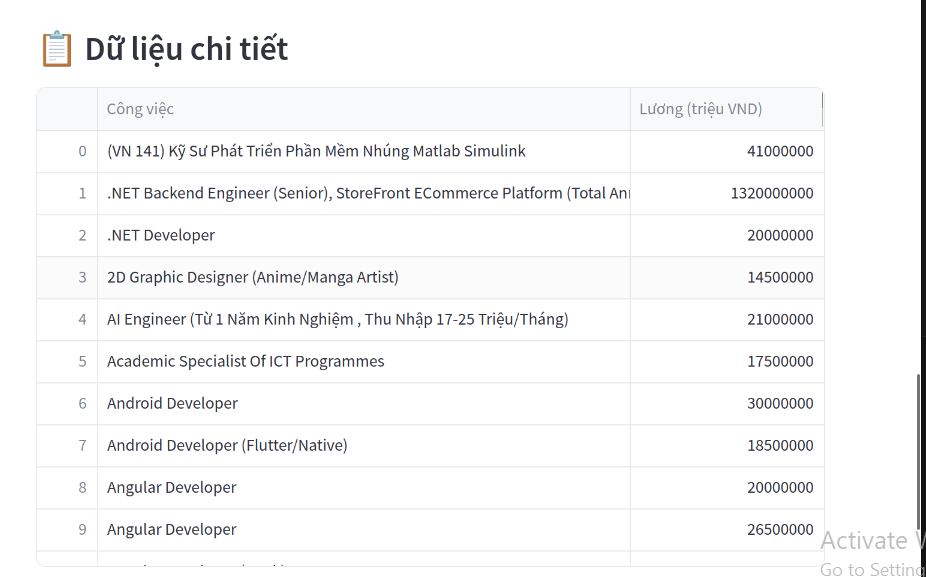
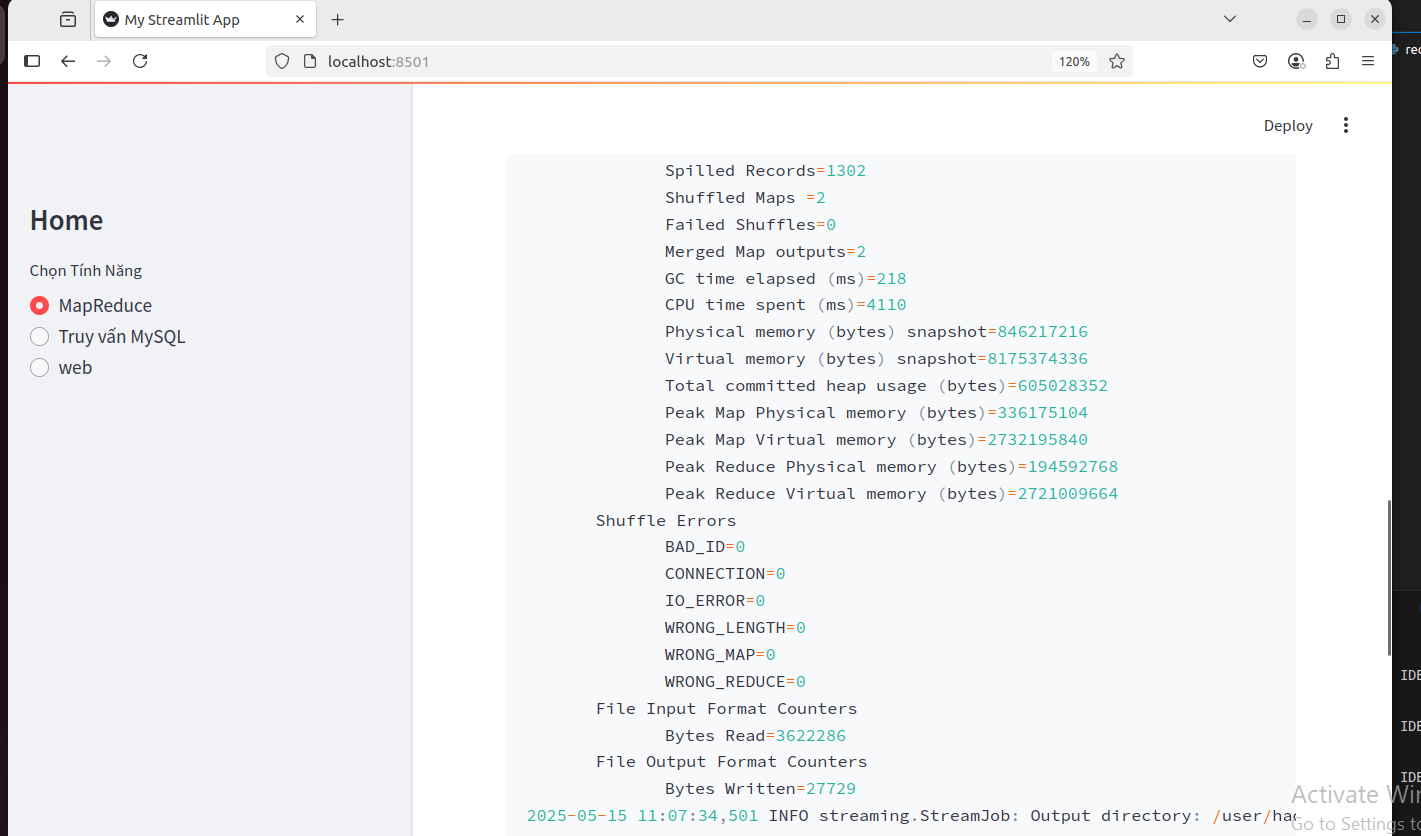
**Xóa Dữ liệu:**



**Hỗ trợ người dùng tương tác UI**

**Hỗ trợ chạy mapreduce**





**Hỗ trợ người dùng tìm kiếm công việc phù hợp**

