

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

LÊ THỊ PHƯƠNG LINH - MSSV: 19127041
BÙI ĐĂNG KHOA - MSSV: 19127645

**HỆ THỐNG GỢI Ý SẢN PHẨM DỰA
TRÊN DỊCH VỤ TƯƠNG TÁC CHO CÁC
DOANH NGHIỆP VỪA VÀ NHỎ**

TÀI LIỆU HƯỚNG DẪN CÀI ĐẶT

THÀNH PHỐ HỒ CHÍ MINH, 2023

Tài liệu này hướng dẫn cách cài đặt và triển khai hệ thống dịch vụ gọi ý tương tác từ thư mục “source” ở hệ điều hành (Ubuntu OS). Source code đã được chúng tôi lưu trên [Github](#).

1. Cài đặt các package sử dụng

Phần này liệt kê các packages cần thiết để xây dựng hệ thống, các packages này là cơ sở để sử dụng các công cụ mà chúng tôi đã trình bày ở báo cáo luận văn.

Package	Mô tả	Phiên bản	Link hướng dẫn
Python	Hệ thống gọi ý sản phẩm được xây dựng trên Django, là một framework web của Python	3.7.9 hoặc 3.8	https://phoenixnap.com/kb/how-to-install-python-3-ubuntu
pip	Trình quản lý cho phép cài đặt các thư viện và module Python một cách dễ dàng	20.1.1	https://phoenixnap.com/kb/how-to-install-pip-on-ubuntu
NodeJS	Giao diện quản lý dữ liệu của hệ thống được xây dựng trên ReactJS, là một framework front-end của Javascript. Để thực thi Javascript, cần cài đặt NodeJS	14.17.3	https://phoenixnap.com/kb/install-latest-node-js-and-nmp-on-ubuntu
npm	Node Package Manager (npm) là trình quản lý gói cho phép cài đặt, quản lý và chia sẻ các gói mã nguồn JavaScript.	8.1.3	https://phoenixnap.com/kb/install-latest-node-js-and-nmp-on-ubuntu
virtuale nv	Là công cụ tạo môi trường ảo Python độc lập, giúp tạo các folders cần thiết để thực thi đồ án	20.16.6	https://packaging.python.org/en/latest/guides/installing-using-pip-and-virtual-environments/

Package	Mô tả	Phiên bản	Link hướng dẫn
Postgres SQL	Hệ quản trị cơ sở dữ liệu quan hệ có mã nguồn mở, giúp quản lý dữ liệu	14.4	(sử dụng neon cloud) https://neon.tech/docs/get-started-with-neon/setting-up-a-project
Airflow	Bộ lập lịch có mã nguồn mở, giúp quản lý các nhiệm vụ chạy theo lịch	2.4.2	https://airflow.apache.org/docs/apache-airflow/stable/installation/installing-from-pypi.html

2. Thiết lập và cấu hình

Sau khi cài đặt đầy đủ các package bên trên, chúng tôi tiến hành các bước thiết lập và cấu hình dựa trên các package trên. Bao gồm các bước: thiết lập cơ sở dữ liệu, kết nối với cơ sở dữ liệu, khởi động recommender server và khởi động giao diện quản trị viên.

a. Thiết lập cơ sở dữ liệu

Tạo cơ sở dữ liệu trên Neon theo đường link hướng dẫn bên trên. Lưu ý ghi chú lại “.env” mà Neon đã cung cấp để có thể kết nối giữa recommender server và Neon một cách dễ dàng.

```
env.txt - Notepad
File Edit Format View Help
# Connection details
PGHOST=
PGDATABASE=neondb
PGUSER=bdkhoa19
PGPASSWORD=sc\

# Connection string
DATABASE_URL=postgres://          .ap-southeast-1.aws.neon.tech/neondb
```

Hình 1. Thiết lập cơ sở dữ liệu

Sau khi lưu được tập tin “.env” này từ trang Neon, có thể coi như ta đã hoàn thành bước thiết lập cơ sở dữ liệu.

b. Thiết lập và cấu hình recommender server

- Bước 1: Chuyển đường dẫn vào thư mục “./recommender-backend-main/”

```
root@Diastic-thesis:~/thesis/trivineu# cd recommender-backend-main/
```

- Bước 2: Tạo môi trường ảo cho recommender server nằm ở trong thư mục “./recommender-backend-main/” có tên django

```
root@Diastic-thesis:~/thesis/trivine# cd recommender-backend-main
root@Diastic-thesis:~/thesis/trivine/recommender-backend-main# python3 -m venv django
```

- Bước 3: install thư viện vào môi trường ảo.
Trước tiên cần kích hoạt sử dụng môi trường ảo.

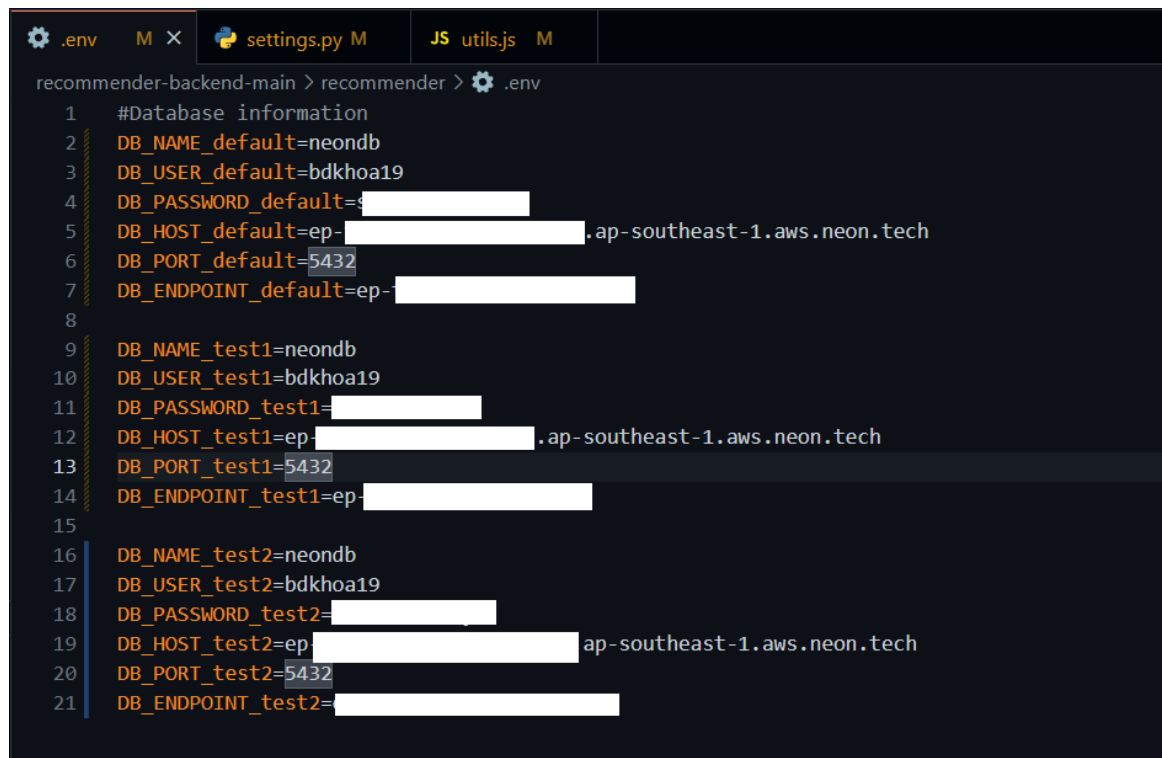
```
root@Diastic-thesis:~/thesis/trivine/recommender-backend-main# source django/bin/activate
```

Sau đó cài đặt các thư viện cần thiết vào môi trường ảo đã chuẩn bị ở file requirement.txt.

```
root@Diastic-thesis:~/thesis/trivine/recommender-backend-main# source django/bin/activate
(django) root@Diastic-thesis:~/thesis/trivine/recommender-backend-main# pip install -r requirements.txt --no-cache-dir
```

Trong trường hợp sử dụng VPS có ít RAM có thể thêm tag `--no-cache-dir` để hạn chế việc tràn RAM khiến không thể cài thư viện.

- Bước 4: Chỉnh file “.env” và file settings



```
recommender-backend-main > recommender > .env
1 #Database information
2 DB_NAME_default=neondb
3 DB_USER_default=bdkhoa19
4 DB_PASSWORD_default=
5 DB_HOST_default=ep- .ap-southeast-1.aws.neon.tech
6 DB_PORT_default=5432
7 DB_ENDPOINT_default=ep-
8
9 DB_NAME_test1=neondb
10 DB_USER_test1=bdkhoa19
11 DB_PASSWORD_test1=
12 DB_HOST_test1=ep- .ap-southeast-1.aws.neon.tech
13 DB_PORT_test1=5432
14 DB_ENDPOINT_test1=ep-
15
16 DB_NAME_test2=neondb
17 DB_USER_test2=bdkhoa19
18 DB_PASSWORD_test2=
19 DB_HOST_test2=ep- .ap-southeast-1.aws.neon.tech
20 DB_PORT_test2=5432
21 DB_ENDPOINT_test2=
```

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': env('DB_NAME_default'),
        'USER': env('DB_USER_default'),
        'PASSWORD': env('DB_PASSWORD_default'),
        'HOST': env('DB_HOST_default'),
        'PORT': env('DB_PORT_default'),
        'OPTIONS': {
            'options': 'project='+env('DB_ENDPOINT_default'),
            'sslmode': 'require'
        },
    },
    'test1': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': env('DB_NAME_test1'),
        'USER': env('DB_USER_test1'),
        'PASSWORD': env('DB_PASSWORD_test1'),
        'HOST': env('DB_HOST_test1'),
        'PORT': env('DB_PORT_test1'),
        'OPTIONS': {
            'options': 'project='+env('DB_ENDPOINT_test1'),
            'sslmode': 'require'
        },
    },
}
```

- Bước 5: Migrate các cơ sở dữ liệu.

```
(django) root@Diasitic-thesis:~/thesis/trivine/recommender-backend-main/recommender# python3 manage.py migrate
```

```
(django) root@Diasitic-thesis:~/thesis/trivine/recommender-backend-main/recommender# python3 manage.py migrate dimadb --database=test1
```

- Bước 6: chạy

```
(django) root@Diasitic-thesis:~/thesis/trivine/recommender-backend-main/recommender# python3 manage.py runserver 104.248.154.31:8000
```

c. Thiết lập và cấu hình react

- Bước 1: chuyển đường dẫn và thư mục

```
(django) root@Diasitic-thesis:~/thesis/trivine# cd recommender-react-main/
```

- Bước 2: npm i

```
(django) root@Diasitic-thesis:~/thesis/trivine/recommender-react-main# npm install
```

- Bước 3: Chỉnh lại constant/utils domainpath

```
.env M settings.py M JS utils.js M X
recommender-react-main > src > constants > JS utils.js > ...
1 export const domainPath = "http://104.248.154.31:8000/"
```

- Bước 4: Chạy

```
(django) root@Diasitic-thesis:~/thesis/trivine# cd recommender-react-main/npm start
```

d. Thiết lập và cấu hình bộ lập lịch airflow

Đưa tập tin “scheduler_job.py” vào thư mục chứa dags của airflow đã thiết lập

chỉnh thành host tương ứng

```
CURRENT_TIME = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
HOST          = 'http://192.168.150.1:8000/'
```

chạy lệnh:

airflow webserver

airflow scheduler

Vào giao diện airflow setup phần connection vào database management



Hình : Tạo connection đến database

Lưu ý: tuy nhiên, trước khi chạy như vậy chúng ta cần mở port

```
(django) root@Diasitic-thesis:~/thesis/trivine# sudo ufw allow 22/tcp
rules updated
```

3. Triển khai

Phần này hướng dẫn cách triển khai hệ thống gợi ý sản phẩm và giao diện quản lý cho admin website thương mại điện tử.

a. Hệ thống gợi ý sản phẩm

Bước	Mô tả	Minh họa
1	Chuyển đường dẫn đến /trivi-backend-main để khởi động môi trường Django	Câu lệnh: django/Scripts/activate

2	Chuyển đường dẫn đến /trivi-backend-main/recommender để chạy ứng dụng	Câu lệnh: python manage.py runserver
3	Mở trình duyệt và dẫn đến địa chỉ của hệ thống gợi ý sản phẩm. Nếu không có lỗi gì xảy ra, màn hình trình duyệt sẽ hiển thị như hình bên.	
4	Đăng nhập vào hệ thống với tài khoản đã được đăng ký với quyền superuser trước đó	

b. Giao diện quản lý

Bước	Mô tả	Minh họa
1	Chuyển đường dẫn đến /trivi-react-main để khởi động giao diện	Câu lệnh: npm start
2	Mở trình duyệt, chọn đến đường dẫn của giao diện, hệ thống sẽ yêu cầu đăng nhập (đối với trường hợp chưa đăng nhập trước đó) hoặc hiển thị giao diện trang chủ (đối với trường hợp đã lưu thông tin đăng nhập)	
3	Tiến hành thao tác trên giao diện	

Các bước thực hiện để nhúng vào trang TMĐT có thể xem ở video demo