**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**PHẠM QUANG MINH**

**<TÊN KHÓA LUẬN TỐT NGHIỆP>**

**ĐỒ ÁN NGÀNH**

**NGÀNH KHOA HỌC MÁY TÍNH**

**TP. HỒ CHÍ MINH, 2021BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**PHẠM QUANG MINH**

**<TÊN KHÓA LUẬN TỐT NGHIỆP>**

**Mã số sinh viên: 1851010075**

**ĐỒ ÁN NGÀNH**

**NGÀNH KHOA HỌC MÁY TÍNH**

**Giảng viên hướng dẫn: LÊ NGỌC HIẾU**

**TP. HỒ CHÍ MINH, 2021**

**Ý KIẾN CHO PHÉP BẢO VỆ ĐỒ ÁN/ KHÓA LUẬN TỐT NGHIỆP**

**CỦA GIẢNG VIÊN HƯỚNG DẪN**

**Giảng viên hướng dẫn:**

**Sinh viên thực hiện: Lớp:**

**Ngày sinh: Nơi sinh:**

**Tên đề tài:**

**Ý kiến của giảng viên hướng dẫn về việc cho phép sinh viên được bảo vệ đồ án/ khóa luận trước Hội đồng:**

*Thành phố Hồ Chí Minh, ngày … tháng … năm ……*

Người nhận xét

…………………………

LỜI CẢM ƠN

Lời đầu tiên xin được cảm ơn các giảng viên tại trường đại học Mở thành phố Hồ Chí Minh đã cùng đồng hành với sinh viên chúng em trong suốt thời gian học tại trường. Cảm ơn các anh chị cùng các bạn tại khoa Công nghệ thông tin đã cùng gắn bó, chia sẻ, dìu dắt để em có thể lựa chọn và tìm ra hướng đi cho riêng mình trong tương lai.

Đặc biệt cảm ơn giảng viên hướng dẫn của em trong đồ án ngành lần này là thầy Lê Ngọc Hiếu đã cùng đồng hành và giúp đỡ em hoàn thành đồ án của mình. Dù gặp nhiều khó khăn khi thực hiện nhưng thầy đã luôn giúp đỡ động viên để có thể hoàn thành đồ án khiến em vô cùng cảm kích.

Đồ án của em lần này còn nhiều thiếu sót nhưng thực sự là những kiến thức em đã tìm hiểu và học hỏi được qua sự giúp đỡ của mọi người. Mong rằng các thầy cô nếu được có thể góp ý giúp em hoàn thiện hơn phục vụ cho công việc sau này.

Một lần nữa xin chân thành cảm ơn tất cả!

*Thành phố Hồ Chí Minh, ngày … tháng … năm ……*

Phạm Quang Minh

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

TÓM TẮT ĐỒ ÁN

Trong đồ án này ta sẽ thực hiện công cụ để chạy được các mã nguồn của các ngôn ngữ lập trình như python, javascript, java từ website. Công cụ này sẽ vận hành bằng cách gửi một yêu cầu lên server có nội dung bao gồm mã nguồn và ngôn ngữ thực hiện sau đó server sẽ trả về kết quả chạy đoạn mã nguồn đó .Để thực hiện được điều này đồ án vận dụng các kiến thức về python django để tạo API, Docker để dùng làm môi trường chạy code từ website, React để tạo giao diện và sử dụng các câu lệnh command line của linux để thực thi code trong môi trường Docker. Kết quả thực hiện là đã chạy được mã nguồn từ website trong Docker container nhưng vẫn còn chậm và phụ thuộc rất nhiều vào tốc độ mạng, từ công cụ này có thể phát triển thêm để trở thành một website học lập trình hiệu quả hoặc một website luyện code, luyện thuật toán mà không cần cài đặt môi trường trên máy tính cá nhân.

**MỤC LỤC**

[LỜI CẢM ƠN 2](#_Toc86548826)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 3](#_Toc86548827)

[TÓM TẮT ĐỒ ÁN 4](#_Toc86548828)

[DANH MỤC TỪ VIẾT TẮT 7](#_Toc86548829)

[DANH MỤC HÌNH VẼ 8](#_Toc86548830)

[DANH MỤC BẢNG 9](#_Toc86548831)

[Chương 1. GIỚI THIỆU ĐỀ TÀI 10](#_Toc86548832)

[1.1. Tổng quan 10](#_Toc86548833)

[1.2. Mục đích 10](#_Toc86548834)

[1.3. Phạm vi và đối tượng nghiên cứu 10](#_Toc86548835)

[Chương 2. NỀN TẢNG LÝ THUYẾT 11](#_Toc86548836)

[2.1. Giới thiệu Docker 11](#_Toc86548837)

[2.1.1. Tổng quan về Docker 11](#_Toc86548838)

[2.1.2. Cơ chế hoạt động 11](#_Toc86548839)

[2.1.3. Kiến trúc của Docker 11](#_Toc86548840)

[2.1.4. Các thành phần chính trong Docker 12](#_Toc86548841)

[2.1.5. Cài đặt và sử dụng Docker 12](#_Toc86548842)

[2.2. Giới thiệu Django và Django REST Framework 14](#_Toc86548843)

[2.2.1. Django là gì? 14](#_Toc86548844)

[2.2.2. Mô hình MVT 14](#_Toc86548845)

[2.2.3. Tổng quan về Django REST Framework 15](#_Toc86548846)

[2.2.4. Tạo project 15](#_Toc86548847)

[2.3. Giới thiệu React 17](#_Toc86548848)

[2.3.1. Tổng quan về React 17](#_Toc86548849)

[Chương 3. HỆ THỐNG ONLINE COMPILER 18](#_Toc86548850)

[3.1. Chủ đề cấp độ 2 18](#_Toc86548851)

[3.1.1. Chủ đề cấp độ 3 18](#_Toc86548852)

[3.2. Chủ đề cấp độ 2 18](#_Toc86548853)

[Chương 4. Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 19](#_Toc86548854)

[4.1. Kết quả đạt được 19](#_Toc86548855)

[4.2. Chức năng cần cải thiện 19](#_Toc86548856)

[4.3. Hướng phát triển 19](#_Toc86548857)

[TÀI LIỆU THAM KHẢO 20](#_Toc86548858)

DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| STT | Từ viết tắt | Giải nghĩa |
| 1 | REST | Representational state transfer |
| 2 | API | Application Programming Interface |
| 3 | MVC | Model – View – Controller |
| 4 | MVT | Model – View – Template |
| 5 | CLI | Command line interface |
| 6 | -t | Tag |
| 7 | -rm | Remove |
| 8 | -rmi | Remove image |

DANH MỤC HÌNH VẼ

Hình 2.2.1: Kiến trúc của Docker (https://docs.docker.com/get-started/overview/) 12

[Hình 2.2: Tạo project Pycharm 16](#_Toc86549509)

DANH MỤC BẢNG

[Bảng 2.1: Mô hình MVC và MVT 16](#_Toc86545172)

# GIỚI THIỆU ĐỀ TÀI

Trong chương này giúp ta nắm được khái quát về đề tài

## Tổng quan

Trình biên dịch trực tuyến thường được sử dụng để chạy các mã nguồn trực tuyến mà không cần phải cài đặt môi trường trên máy, đề tài dựa trên các kiến thức về docker, django và react để tạo ra một trình biên dịch trực tuyến từ đó có thể phát triển thêm để sử dụng vào ứng dụng web để học lập trình trực tuyến hoặc code challenge

## Mục đích

Đề tài sẽ tập trung hoàn thiện việc thực hiện trình biên dịch trực tuyến cho đa ngôn ngữ như java, python, javascript để giúp người sử dụng có thể chạy các dòng code đơn giản mà không cần cài đặt các môi trường trên máy cũng như từ công cụ này có thể phát triển lên một website học tập hoặc code challenge.

## Phạm vi và đối tượng nghiên cứu

Để tạo ra một công cụ biên dịch trực tuyến, đề tài tìm hiểu sâu về các khái niệm, cách thức hoạt động, tổ chức của Python Django Rest Framework, React, Docker.

# NỀN TẢNG LÝ THUYẾT

Chương này giúp ta làm rõ, nắm được các khái niệm và một số các câu lệnh cơ bản được sử dụng để xây dựng ứng dụng

## Giới thiệu Docker

### Tổng quan về Docker

Docker là một nền tảng mã nguồn mở được dùng để phát triển, triển khai, thực thi ứng dụng. Docker giúp tách biệt các ứng dụng khỏi môi trường máy tính phát triển để có thể phân phối phần mềm, ứng dụng một cách nhanh chóng.

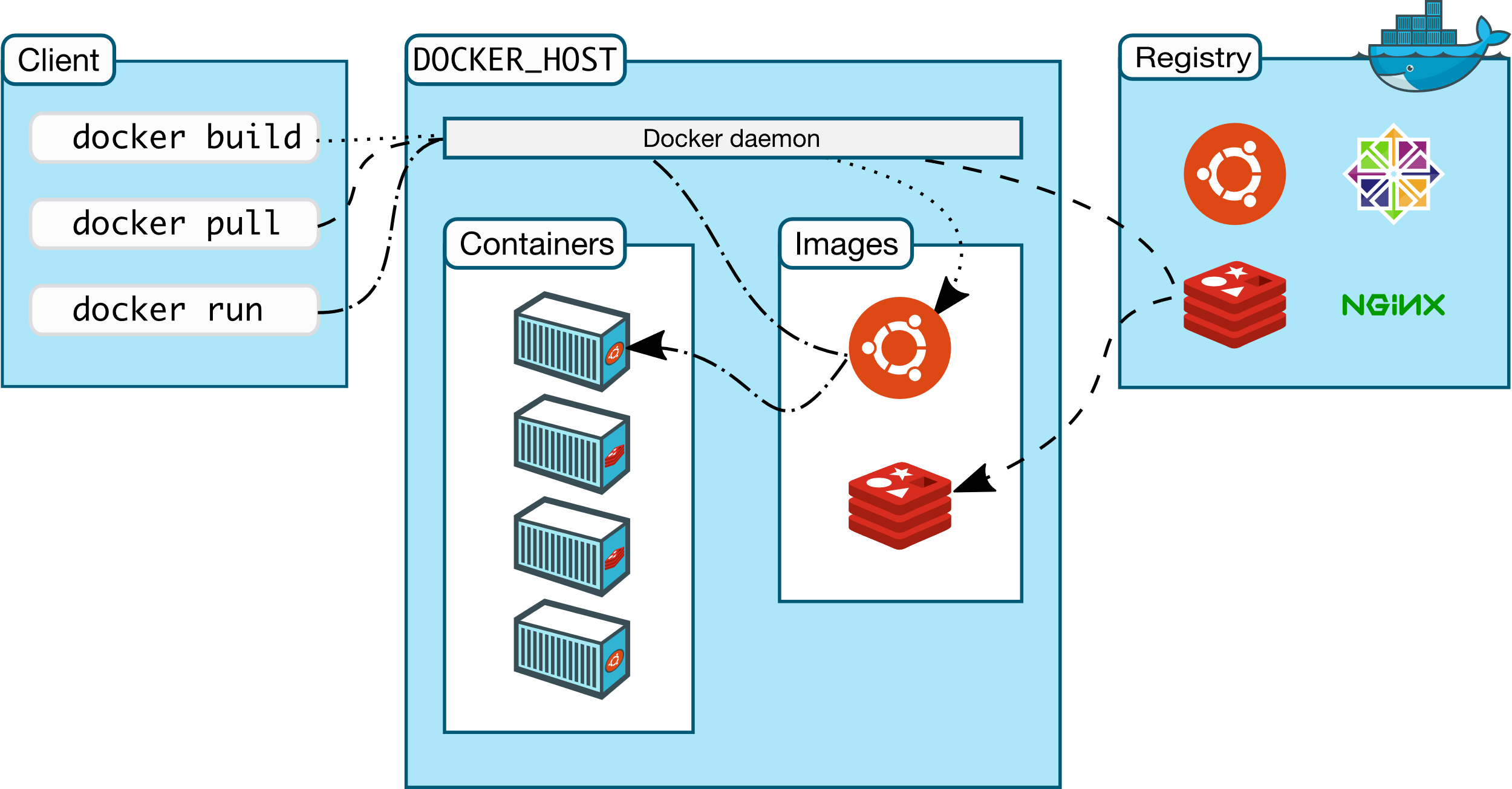
### Cơ chế hoạt động

Docker cung cấp khả năng đóng gói và chạy ứng dụng trong một môi trường bị cô lập được gọi là container. Sự cô lập và bảo mật cho phép ta có thể chạy nhiều conainer đồng thời trên cùng một máy chủ. Các container có dung lượng nhẹ và chứa mọi thứ cần thiết để chạy ứng dụng bên trong nó, chính vì thế ta không cần phải dựa vào những gì được cài đặt trên máy chủ để chạy phần mềm. Nhờ vậy nên ta có thể dễ dàng chia sẻ các container và chạy chúng ở các máy chủ khác mà không cần cài đặt các công cụ mà ứng dụng yêu cầu lên máy chủ.

Docker được tạo ra để sử dụng trên nền tảng Linux nhưng hiện nay đã được phát triển để chạy trên các nền tảng khác như Windows, MacOS.

### Kiến trúc của Docker

Docker sử dụng kiến trúc client – server. Docker client sẽ giao tiếp với Docker daemon là nơi thực hiện các công việc xây dựng, chạy và phân phối các Docker container. Docker client và daemon có thể chạy trên cùng một hệ thống hoặc kết nối từ xa. Hai bên giao tiếp với nhau bằng cách sử dụng REST API, Unix socket hoặc kết nối mạng. Một dạng Docker client khác là Docker Compose, cho phép ta làm việc với các ứng dụng trên một tập các container.



Hình 2.1: Kiến trúc của Docker (<https://docs.docker.com/get-started/overview/>)

### Các thành phần chính trong Docker

Khi sử dụng Docker, ta tạo ra và sử dụng các image, container, mạng, volume, plugin, và các thành phần khác. Ở đây ta cần nắm một số các thành phần:

* Image là các mẫu chỉ đọc với các hướng dẫn để tạo ra Docker container. Ta có thể tự tạo một image hoặc sử dụng image do người khác tạo và được đưa lên registry.
* Dockerfile là một file chứa các lệnh để xây dựng các Docker image.
* Container là nơi thực thi các thể hiện của một image. Ta có thể tạo, chạy, di chuyển hoặc xóa các container bằng Docker API hoặc CLI. Ta có thể kết nối container với một hoặc nhiều mạng, đính kèm bộ nhớ hoặc tạo một image dựa trên trạng thái hiện tại của container.

### Cài đặt và sử dụng Docker

Để sử dụng Docker và các dịch vụ của nó ta cần cài đặt phiên bản đầy đủ của Docker.

Đối với Windows hoặc MacOS ta có thể tải và cài đặt phiên bản mới nhất tại <https://docs.docker.com/desktop/>.

Còn đối với hệ điều hành linux ta có thể xem cụ thể cách cài đặt trên từng phiên bản hệ điều hành tại <https://docs.docker.com/engine/install/>.

Sau khi đã cài đặt được Docker ta có thể khởi chạy container đầu tiên trên Docker bằng cách mở command line và gõ lệnh sau

docker run hello-world

Để tạo Docker image ta cần viết Dockerfile sau đó chạy lệnh cli như sau

docker build [OPTIONS] [path to Dockerfile]

Để chạy Docker image

docker run [OPTIONS] [image tag]

Các OPTIONS để tạo Docker image có thể xem đầy đủ tại <https://docs.docker.com/engine/reference/commandline/build/>

Trong đồ án này ta sẽ chỉ sử dụng một số các OPTIONS như -t, -rm, -rmi.

Cụ thể từ đồ án này ta cần tạo các image chứa và chạy các đoạn code của người dùng viết và gửi từ website. Đoạn code gửi về đơn giản là in ra dòng chữ “Hello world!!” trên ngôn ngữ python thì ta cần một tập tin python có chứa đoạn code và một Dockerfile như sau

src

|---app

|---Dockerfile

|---main.py

Nội dung của Dockerfile

FROM python:3

WORKDIR /usr/src/app

COPY . .

CMD [ "python", "./main.py" ]

Sau đó ta bắt đầu tạo ra Docker image từ Dockerfile

docker build -t pythonscript ./src/app

Để khởi chạy Docker image vừa được tạo ta sử dụng câu lệnh docker run

docker run pythonscript

Trên đây là các bước cơ bản để khởi tạo Docker image và chạy một Docker container từ image được tạo đơn giản.

## Giới thiệu Django và Django REST Framework

### Django là gì?

Django là một web framework mã nguồn mở được viết bằng python và phát hành vào năm 2005 bởi Django Software Foundation. Mục tiêu của django là giúp đơn giản hóa việc phát triển website bằng python

### Mô hình MVT

Django sử dụng mô hình được gọi là MVT (Model – View – Template). Thực tế đây là một biến thể từ mô hình MVC (Model – View – Controller) và cũng chia ứng dụng ra thành 3 thành phần để xử lý: phần model là nơi chứa các thực thể để tương tác với cơ sở dữ liệu; phần view là nơi xử lý các logic tương đương với phần controller trong MVC; phần template là nơi hiển thị ra giao diện cho người dùng tương đương với view trong MVC.

|  |  |  |
| --- | --- | --- |
| Mô hình MVC | Mô hình MVT | Chức năng |
| Model | Model | Tương tác với cơ sở dữ liệu |
| View | Template | Hiển thị giao diện |
| Controller | View | Xử lý logic |

Bảng 2.1: Mô hình MVC và MVT

### Tổng quan về Django REST Framework

#### API

API là viết tắt của Application Programming Interface, là tập hợp các quy tắc và cơ chế mà thông qua đó giúp các ứng dụng và các thành phần tương tác với nhau hay có thể nói các ứng dụng hoặc các thành phần tương tác với nhau thông qua API.

#### REST và RESTful API

REST là viết tắt của Representational state transfer, là một kiểu kiến trúc phần mềm được tạo ra để thiết kế và phát triển ứng dụng sử dụng mạng. Nó là tập hợp các ràng buộc được sử dụng cho việc triển khai các phương tiện internet như web.

REST thường được sử dụng trong việc phát triển API. Một API tuân theo các ràng buộc của REST được gọi là RESTful API, các API này sử dụng phương thức HTTP đơn giản để kết nối đến các tài nguyên giữa các máy thông qua url có các tham số được mã hóa và dùng Json hoặc xml để truyền dữ liệu.

#### Django REST Framework

Django REST Framework là một framework được phát triển để giúp xây dựng các RESTful API trong python django một cách đơn giản và thuận tiện

### Tạo project

Để tạo project python có sử dụng django ta cần tạo một project python thông thường và cài đặt django.

Ở đồ án này ta sẽ sử dụng Pycharm để viết phần python django. Trước tiên ta tạo project trong Pycharm

Graphical user interface, text, application, email

Description automatically generated

Hình .: Tạo project Pycharm

Hình 2.2: Tạo project Pycharm

Sau đó để cài đặt django ta chạy lệnh pip trong terminal

pip install django

Để khởi tạo django project ta thực hiện lệnh trong terminal

django-admin startproject myproject

Sau khi chạy xong lệnh khởi tạo django project ta sẽ có được thư mục của project như sau

new\_project

|---myproject

|---myproject

|---\_init\_.py

|---asgi.py

|---settings.py

|---urls.py

|---wsgi.py

|---manage.py

Để chạy server ta vào thư mục của project vừa tạo và trên cửa sổ terminal gõ lệnh

python manage.py runserver

Server sẽ được khởi chạy mặc định tại port 8000

## Giới thiệu React

### Tổng quan về React

React là một thư viện front-end mã nguồn mở được viết bằng Javascript để xây dựng giao diện người dùng. Được duy trì và phát triển bởi facebook cùng cộng đồng các lập trình viên, React được sử dụng để xây dựng các website dạng single-page hoặc các ứng dụng di động.

React chia các thành phần trên giao diện người dùng phức tạp thành các thành phần nhỏ, độc lập gọi là component. Nhờ đó mà việc thiết kế và phát triển giao diện người dùng trở nên đơn giản hơn nhiều.

# HỆ THỐNG ONLINE COMPILER

## Chủ đề cấp độ 2

Nội dung …………………

Nội dung………………….

### Chủ đề cấp độ 3

#### Chủ đề cấp độ 4

## Chủ đề cấp độ 2

# Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết quả đạt được

## Chức năng cần cải thiện

## Hướng phát triển

TÀI LIỆU THAM KHẢO

(Theo chuẩn IEEE)

“Docker Document,” [Online]. Available: <https://docs.docker.com/>

Wikipedia, “Docker,” 10 10 2021. [Online]. Available: https://vi.wikipedia.org/wiki/Docker\_(phần\_mềm)

“Docker,” [Online]. Available: <https://searchitoperations.techtarget.com/definition/Docker>