

Trường Đại Học Sư Phạm Kỹ Thuật TP. Hồ Chí Minh

Khoa Công Nghệ Thông Tin



BÁO CÁO
ĐIỆN TOÁN Đám Mây

ĐỀ TÀI:
WEB QUẢN LÝ SINH VIÊN KẾT NỐI NHIỀU
DOCKER VỚI NHAU

Nhóm sinh viên thực hiện:

Lê Chí Hiếu **18133012**

Đỗ Đình Phùng **18133040**

GVHD: T.S Huỳnh Xuân Phụng

Tp. Hồ Chí Minh, 26/05/2021

ĐIỂM SỐ

TIÊU CHÍ	NỘI DUNG	TRÌNH BÀY	TỔNG
ĐIỂM			

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Đánh giá.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Giảng viên hướng dẫn

(ký và ghi rõ họ tên)

LỜI CẢM ƠN

Để hoàn thành tốt đề tài và bài báo cáo này, chúng em xin gửi lời cảm ơn chân thành đến giảng viên Huỳnh Xuân Phụng, người đã trực tiếp hỗ trợ chúng em trong suốt quá trình làm đề tài. Chúng em cảm thấy thầy đã đưa ra những lời khuyên từ kinh nghiệm thực tiễn của mình để định hướng cho chúng em đi đúng với yêu cầu của đề tài đã chọn, luôn giải đáp thắc mắc và đưa ra những góp ý, chỉnh sửa kịp thời giúp chúng em khắc phục nhược điểm và hoàn thành tốt cũng như đúng thời hạn Khoa đã đề ra. Chúng em cũng xin gửi lời cảm ơn chân thành các quý thầy cô trong khoa Công Nghệ Thông Tin đã tận tình truyền đạt những kiến thức cần thiết giúp chúng em có nền tảng để làm nên đề tài này, đã tạo điều kiện để chúng em có thể tìm hiểu và thực hiện tốt đề tài. Cùng với đó, chúng em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp chúng em có thể hoàn thiện hơn đề tài của mình. Đề tài và bài báo cáo được chúng em thực hiện trong khoảng thời gian ngắn, với những kiến thức còn hạn chế cùng nhiều hạn chế khác về mặt kĩ thuật và kinh nghiệm trong việc thực hiện một dự án phần mềm. Do đó, trong quá trình làm nên đề tài có những thiếu sót là điều không thể tránh khỏi nên chúng em rất mong nhận được những ý kiến đóng góp quý báu của các quý thầy cô để kiến thức của chúng em được hoàn thiện hơn và chúng em có thể làm tốt hơn nữa trong những lần sau.

TP.HCM, ngày 26 tháng 05 năm 2021

Nhóm sinh viên thực hiện

MỤC LỤC

MỤC LỤC BẢNG	5
MỤC LỤC HÌNH.....	6
Chương 1: Tổng quan chương trình	1
1. Giới thiệu đề tài	1
2. Tìm hiểu về docker	1
2.1. Docker là gì?.....	1
2.2. Các khái niệm liên quan.....	1
2.3. Một số lệnh cơ bản.....	2
Chương 2: Kế hoạch thực hiện	4
1. Kế hoạch	4
2. Phân công công việc	4
Chương 3: Demo ứng dụng.....	5
1. Công nghệ sử dụng	5
1.1. Frontend	5
1.2. Backend	7
1.3. Database.....	12
2. Giao diện web.....	12
3. Dockerfile và docker compose của project	13
3.1. Dockerfile backend	14
3.2. Dockerfile frontend.....	14
3.3. Docker compose.	15
4. Hướng dẫn sử dụng	17
Chương 5: Kết luận và hướng phát triển	21
1. Kết luận	21
1.1. Ưu điểm	21
1.2. Nhược điểm	21
2. Hướng phát triển	21
Tài liệu tham khảo	22

MỤC LỤC BẢNG

Table 1: Kế hoạch theo tuần.....	4
Table 2:Phân công công việc & đóng góp của mỗi sinh viên.....	4

MỤC LỤC HÌNH

Hình 1: Xây dựng frontend cho trang web	5
Hình 2: Thực thi frontend trên visual studio code	6
Hình 3: Màn hình giao diện web	6
Hình 4: Cài đặt các gói của Spring-boot	7
Hình 5: Tiến hành cài đặt các gói Spring-boot vào eclipse.....	8
Hình 6: Tạo Spring Starter Project	8
Hình 7: Điền các thông tin cần thiết cho project	9
Hình 8: Kết nối đến database mysql	10
Hình 9: Thực thi project với maven	10
Hình 10: Nén project thành file jar.....	11
Hình 11: Kết quả sau khi build được file jar	12
Hình 12: Giao diện chức năng thêm sinh viên	13
Hình 13: Giao diện chức năng update sinh viên.....	13
Hình 14: giao diện chức năng danh sách sinh viên sau khi thêm thành công	13
Hình 15: Docker file backend	14
Hình 16: Docker file frontend	15
Hình 17: Docker compose	16
Hình 18: Hướng dẫn sử dụng chạy docker-compose.yml.....	18
Hình 19: Chạy docker-compose.yml trên terminal	18
Hình 20: Các images sau khi build thành công	19
Hình 21: Các container sau khi build thành công	19

Chương 1: Tổng quan chương trình

1. Giới thiệu đề tài

Thời trước đây để xây dựng một ứng dụng hay một trang web cho khách hàng thì buộc khách hàng phải có các phần mềm lập trình hay nền tảng lập trình để chạy được phần mềm ứng dụng. Hiện nay thì không cần phải như thế chỉ cần người lập trình xây dựng các ứng dụng phần mềm hay trang web và xây dựng thành một file. Và khách hàng chỉ việc chạy file này thì có thể sử dụng ứng dụng hay trang web mà không cần phải tải các phần mềm lập trình. File này được gọi là docker-compose.yml. Đề tài nhóm chọn quản lý sinh viên với các chức năng cơ bản như thêm sinh viên, sửa sinh viên và xóa sinh viên. Xây dựng ba thành phần đó là frontend, backend và database sau đó xây dựng ba thành phần này tạo thành các container. Sau đó build các container này lên docker hub để lưu trữ (giống như git) và sử dụng file docker-compose.yml để chạy các container này.

2. Tìm hiểu về docker

2.1. Docker là gì?

- Docker là nền tảng phần mềm cho phép bạn dựng, kiểm thử và triển khai ứng dụng một cách nhanh chóng.
- Docker đóng gói phần mềm vào các đơn vị tiêu chuẩn hóa được gọi là container có mọi thứ mà phần mềm cần để chạy, trong đó có thư viện, công cụ hệ thống, mã và thời gian chạy.

2.2. Các khái niệm liên quan

- Docker Engine: là thành phần chính của Docker, như một công cụ để đóng gói ứng dụng
- Docker Hub: là một “github for docker images”. Trên DockerHub có hàng ngàn public images được tạo bởi cộng đồng cho phép bạn dễ dàng tìm thấy những image mà bạn cần. Và chỉ cần pull về và sử dụng với một số config mà bạn mong muốn.

- Container: là một instance của một image. Bạn có thể create, start, stop, move or delete container dựa trên Docker API hoặc Docker CLI.
- Images: là một khuôn mẫu để tạo một container. Thường thì image sẽ dựa trên 1 image có sẵn với những tùy chỉnh thêm. Ví dụ bạn build 1 image dựa trên image Centos mẫu có sẵn để chạy Nginx và những tùy chỉnh, cấu hình để ứng dụng web của bạn có thể chạy được. Bạn có thể tự build một image riêng cho mình hoặc sử dụng những image được chia sẻ từ cộng đồng Docker Hub. Một image sẽ được build dựa trên những chỉ dẫn của Dockerfile.
- Docker Client: là một công cụ giúp người dùng giao tiếp với Docker host.
- Docker Daemon: lắng nghe các yêu cầu từ Docker Client để quản lý các đối tượng như Container, Image, Network và Volumes thông qua REST API. Các Docker Daemon cũng giao tiếp với nhau để quản lý các Docker Service.
- Dockerfile: là một tập tin bao gồm các chỉ dẫn để build một image.
- Volumes: là phần dữ liệu được tạo ra khi container được khởi tạo.

2.3. Một số lệnh cơ bản

- docker --version: kiểm tra phiên bản docker
- docker info: thông tin hệ thống docker
- docker images -a: liệt kê các image
- docker pull name:image:tag: tải về một image từ hub.docker.com
- docker ps: liệt kê các container đang chạy
- docker ps -a: liệt kê các container
- docker container ls -a: liệt kê các container
- docker run -it --name "nameyourcontainer" -h "nameyourhost" image_id
tạo, chạy một container từ image với id (name) là image_id.
- docker exec -it container_id command: chạy một lệnh command trên container đang hoạt động
- docker stop container_id: dừng hoạt động một container
- docker start -i container_id: chạy một container

- `docker restart container_id`: khởi động lại container
- `docker rm containerid`: xóa container
- `docker commit container_id imagename:imageversion`: lưu một container đang dừng thành Image

Chương 2: Kế hoạch thực hiện

1. Kế hoạch

Table 1: Kế hoạch theo tuần

Tuần	Công việc
9,10	Tìm hiểu về đề tài và lên kế hoạch thực hiện đề tài.
11,12	Tìm hiểu các thành phần trong dockerfile, docker-compose. Và xây dựng frontend
13	Xây dựng backend và database
14	Kiểm thử, viết báo cáo, slide và củng cố lại kiến thức.

2. Phân công công việc

Table 2: Phân công công việc & đóng góp của mỗi sinh viên

STT	Tên sinh viên	Mô tả công việc	Đóng góp
2	Lê Chí Hiếu	<ul style="list-style-type: none">- Xây dựng kế hoạch thực hiện đề tài- Tìm hiểu hoạt động dockerfile, docker-compose.- Viết backend và database- Viết báo cáo	50%
3	Đỗ Đình Phùng	<ul style="list-style-type: none">- Tìm hiểu hoạt động dockerfile, docker-compose- Viết frontend- Viết báo cáo, slide	50%

Chương 3: Demo ứng dụng

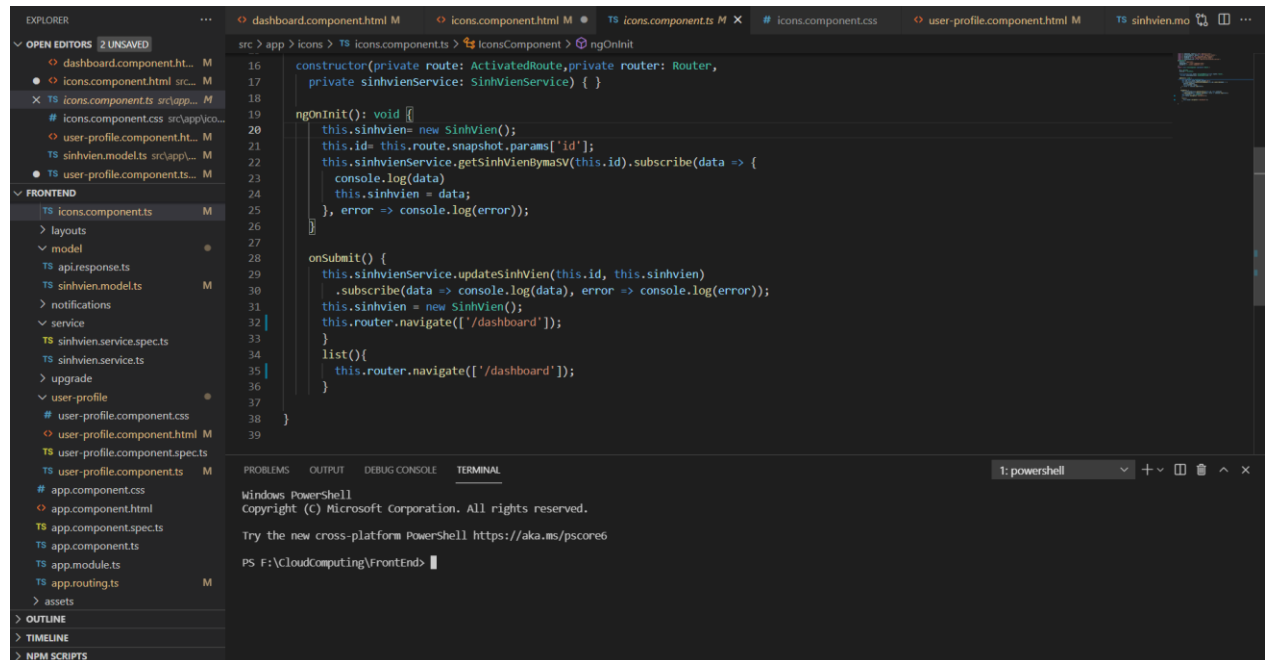
1. Công nghệ sử dụng

Ứng dụng web quản lý sinh viên xây dựng trên ba thành phần chính là:

- Frontend: Sử dụng Angular và bootstrap 5.
- Backend: Sử dụng java Spring-boot
- Database: MySQL

1.1. Frontend

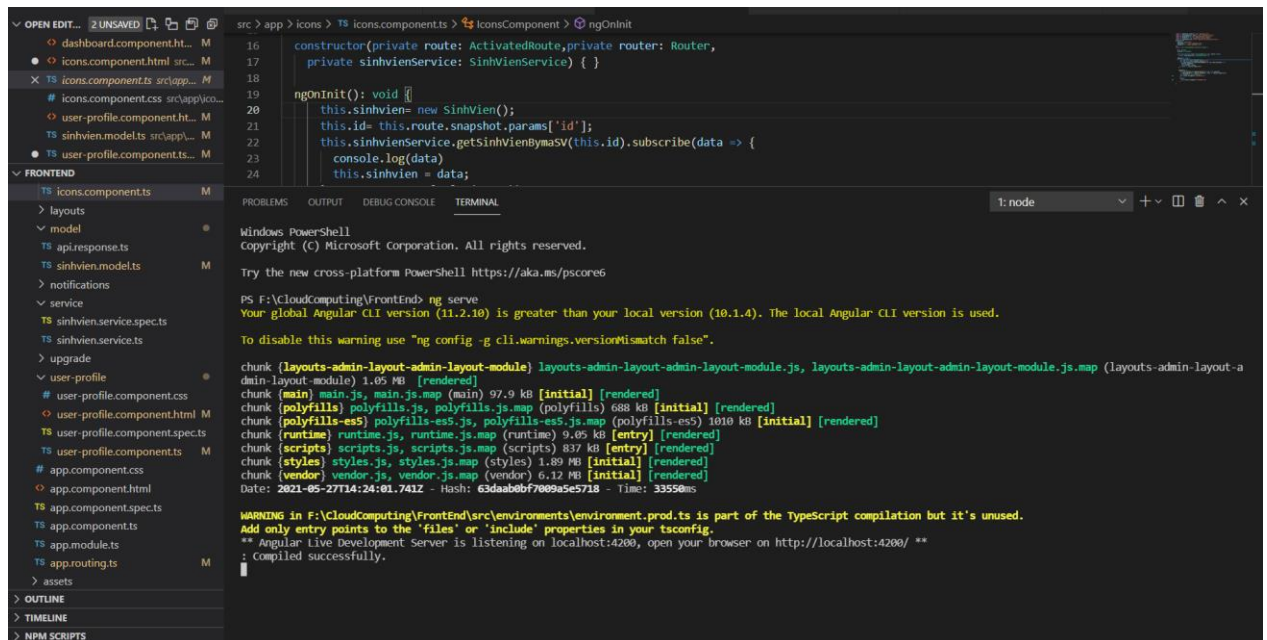
Sử dụng Angular và Bootstrap 5 để xây dựng frontend cho trang web



Hình 1: Xây dựng frontend cho trang web

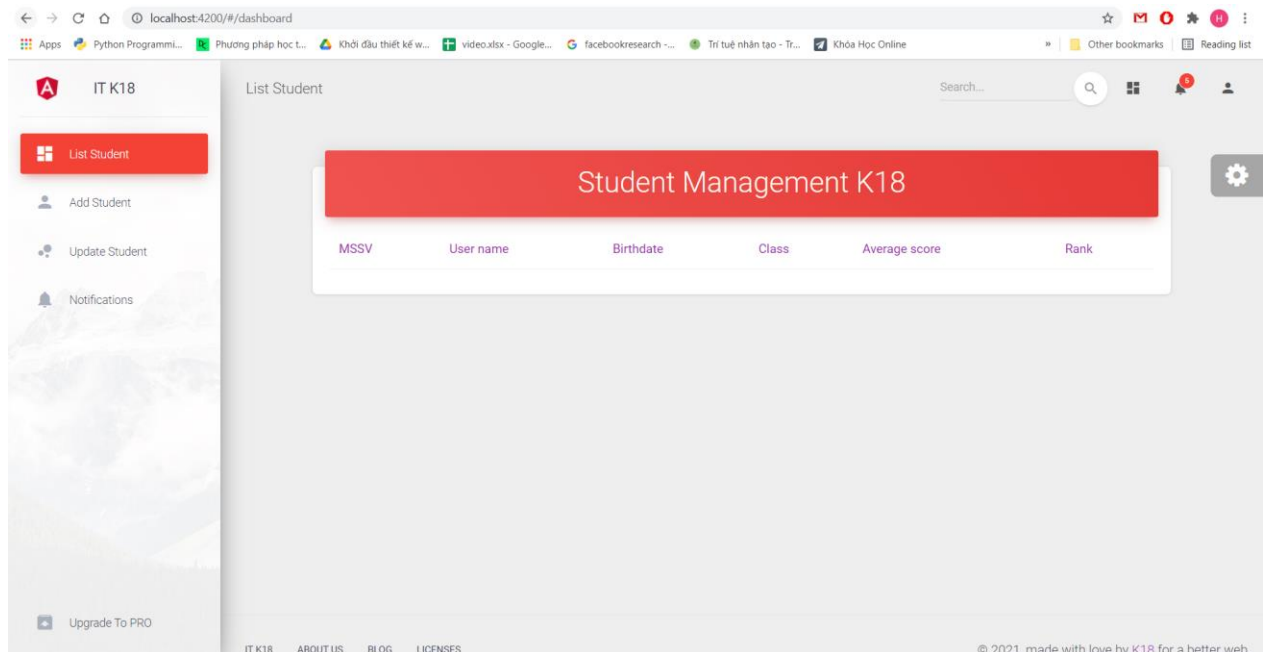
Để chạy frontend trên visual studio code ta chạy: ng serve.

Thì hệ thống tệp sẽ build các gói thư viện cũng như bootstrap.



Hình 2: Thực thi frontend trên visual studio code

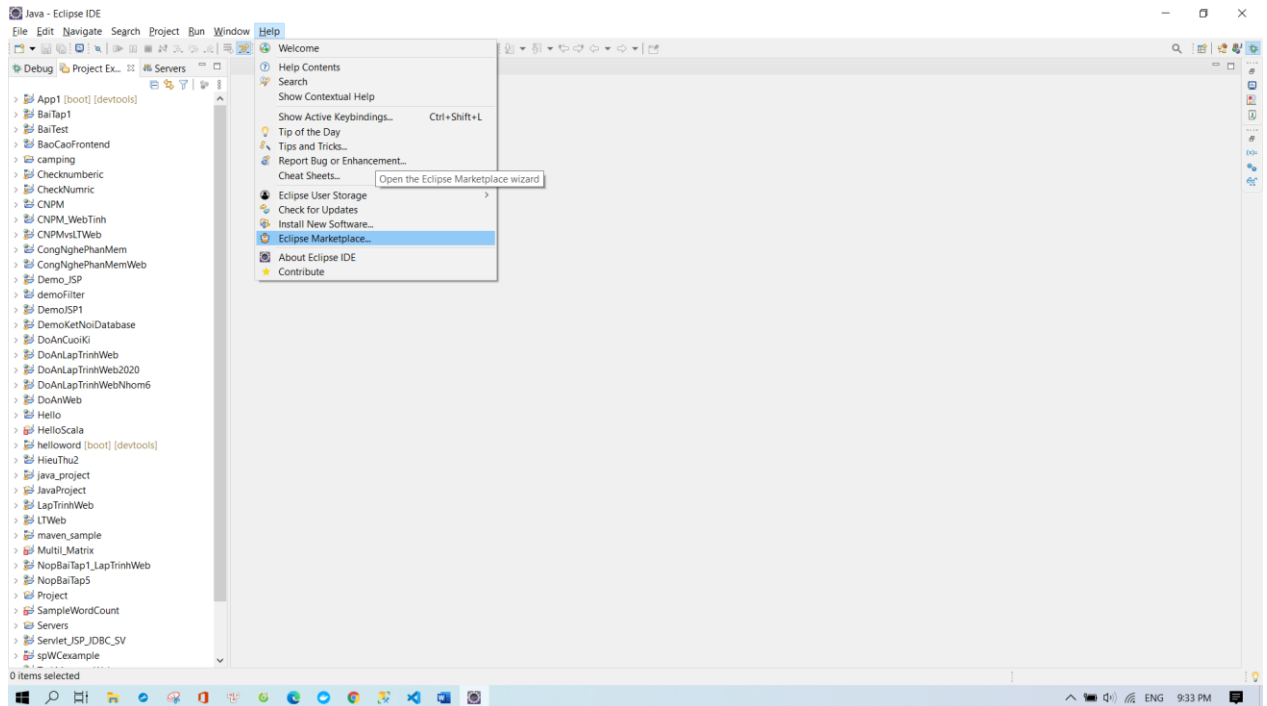
Màn hình giao diện web



Hình 3: Màn hình giao diện web

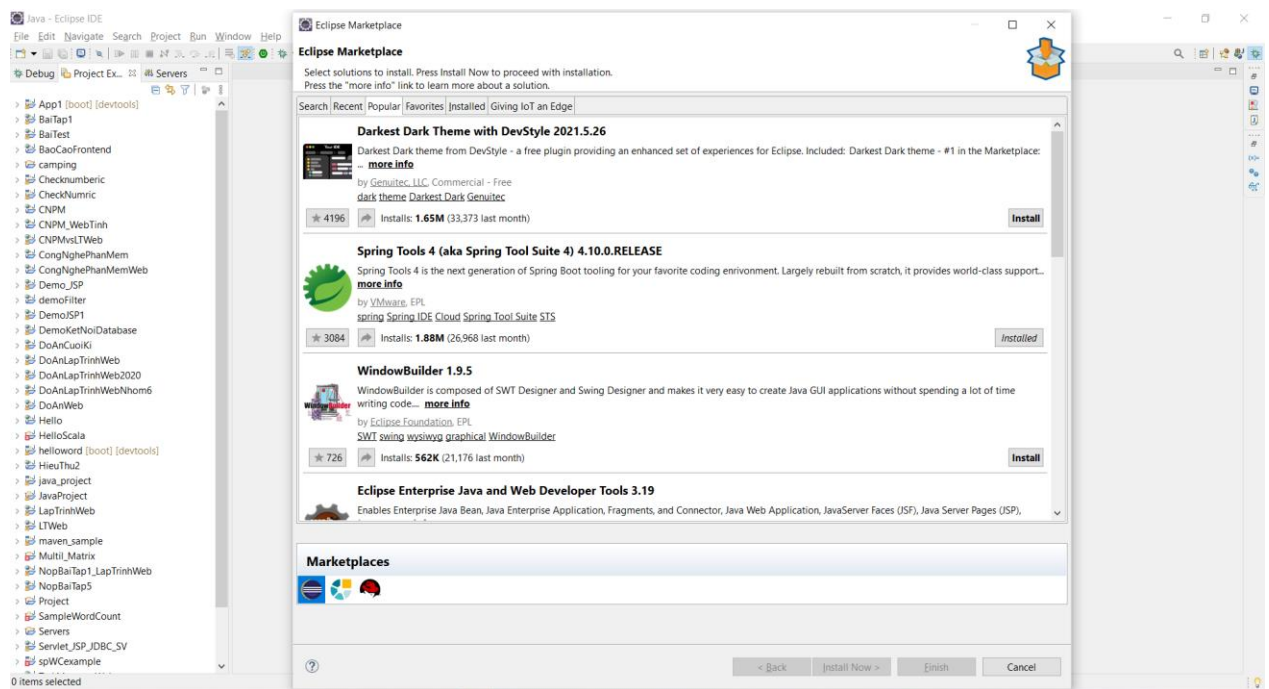
1.2. Backend

Sử dụng java String-boot. Để sử dụng được công nghệ này ta tiến hành cài đặt các gói của String-boot như sau.



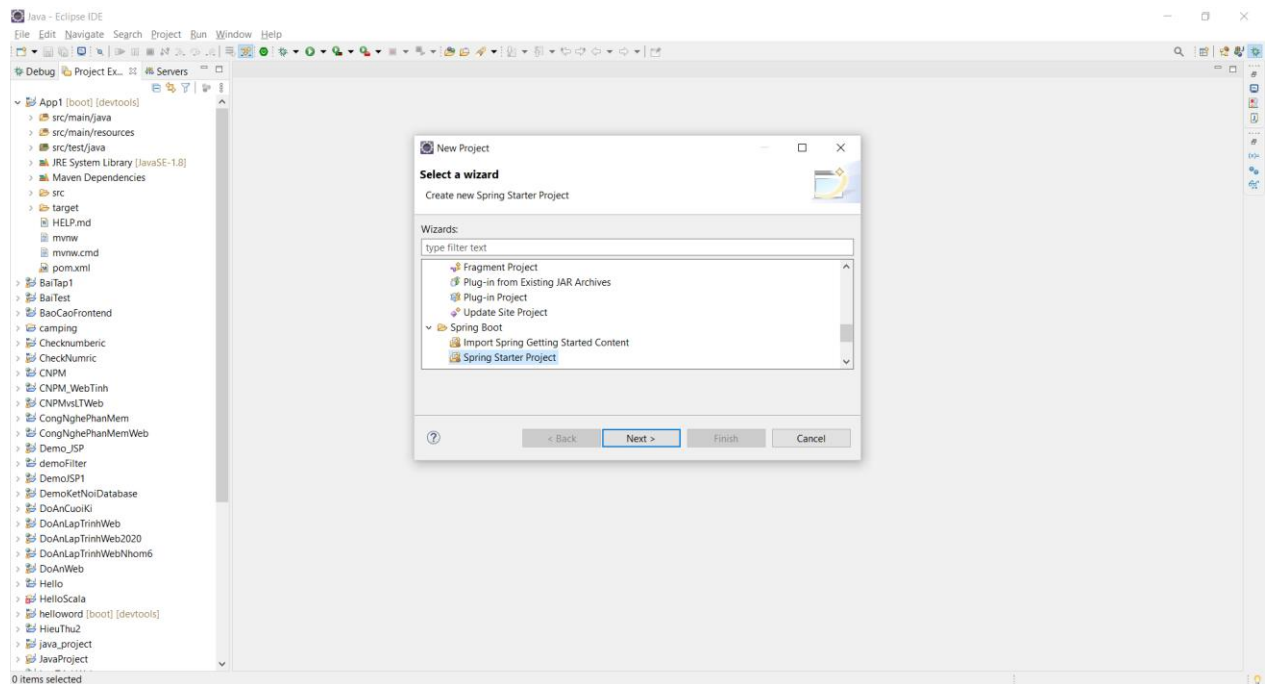
Hình 4: Cài đặt các gói của String-boot

Sau đó ta gõ string tools 4 và tiến hành cài đặt gói vào eclipse.



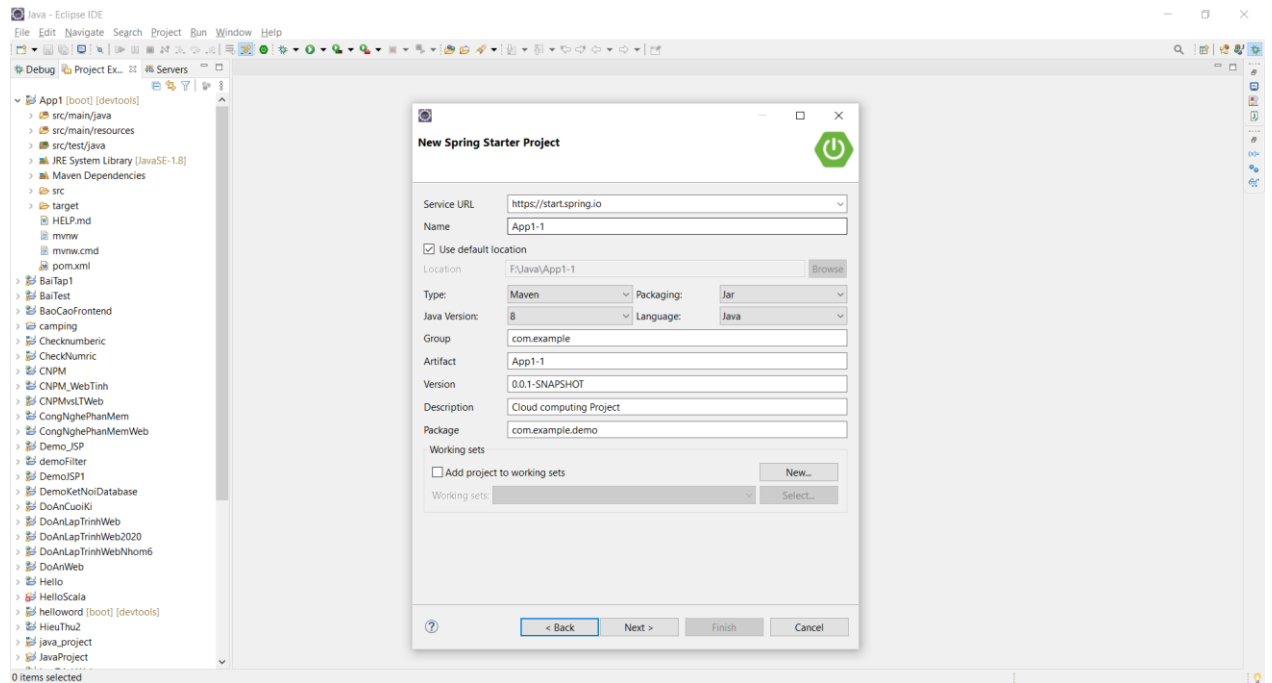
Hình 5: Tiến hành cài đặt các gói String-boot vào eclipse

Sau khi cài đặt xong ta tạo project string



Hình 6: Tạo Spring Starter Project

Sau đó điền các thông tin vào như hình

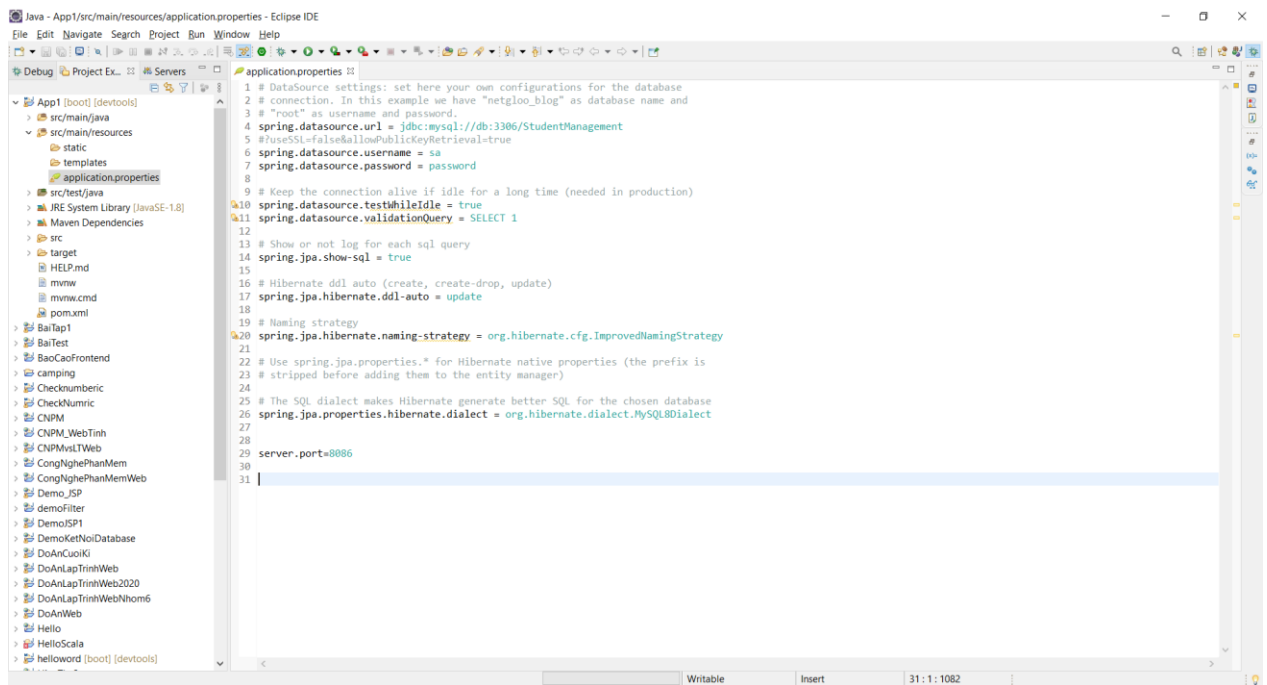


Hình 7: Điền các thông tin cần thiết cho project

Như vậy đã tạo xong project string-boot.

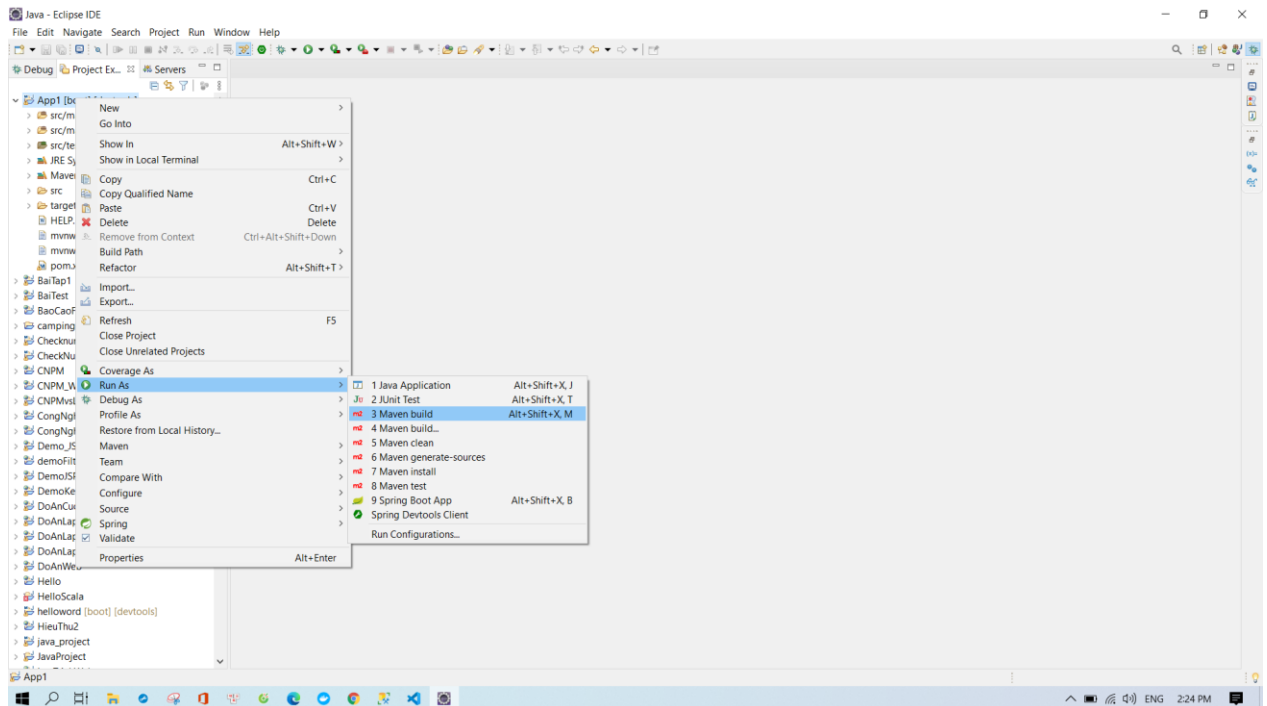
Các thành phần kết nối với cơ sở dữ liệu như sau:

Chỉ ra đường dẫn tới mysql thông qua jdbc:mysql://db:3306/StudentManagement. Ở đây db:3306 là database của mysql luôn luôn có port là 3306 và StudentManagement là tên của database. Chỉ ra được username và password của database mysql để kết nối tới.



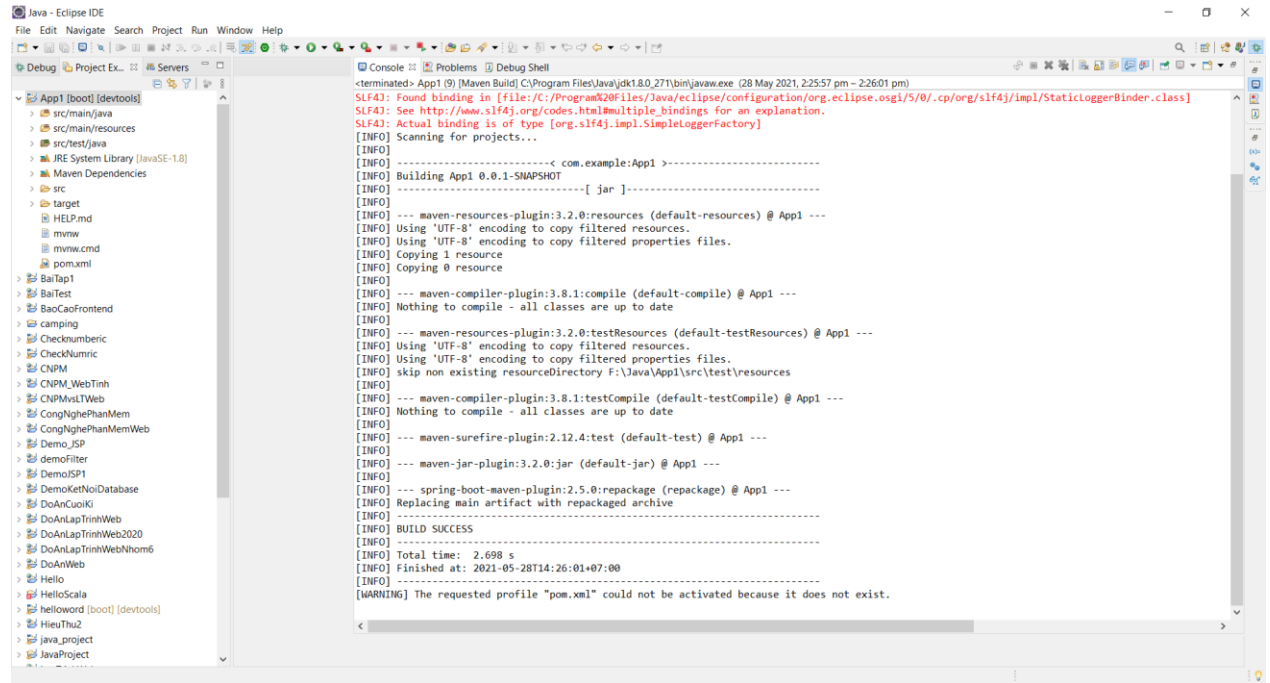
Hình 8: Kết nối đến database mysql

Chạy project với maven build.



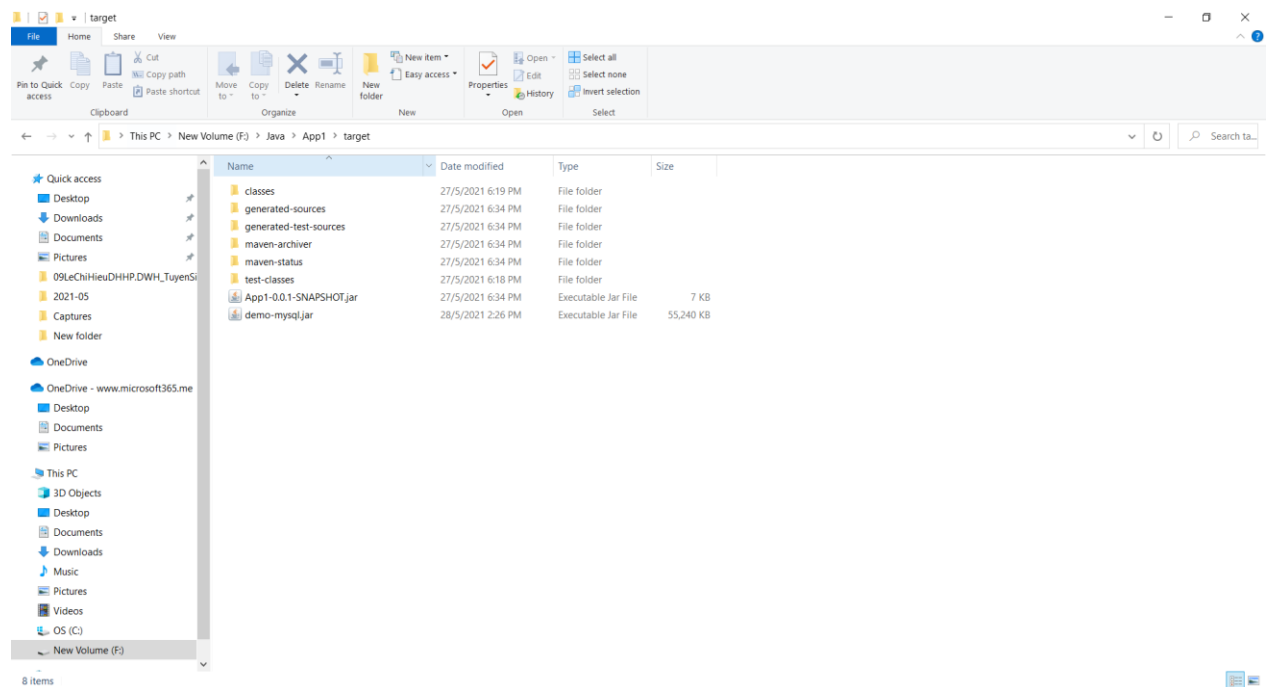
Hình 9: Thực thi project với maven

Nén project thành file jar. File jar khi ta viết dockerfile ghi để build project backend.



Hình 10: Nén project thành file jar

Sau khi build thành công ta được file jar tên demo-mysql.jar.



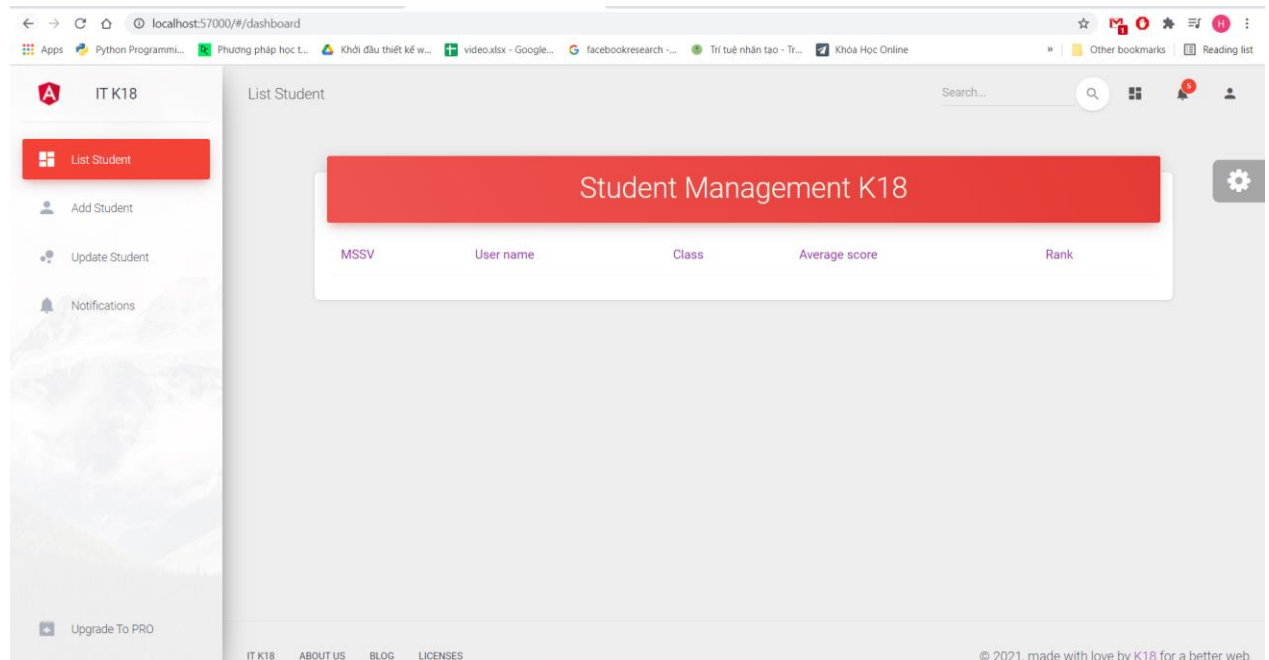
Hình 11: Kết quả sau khi build được file jar

1.3. Database

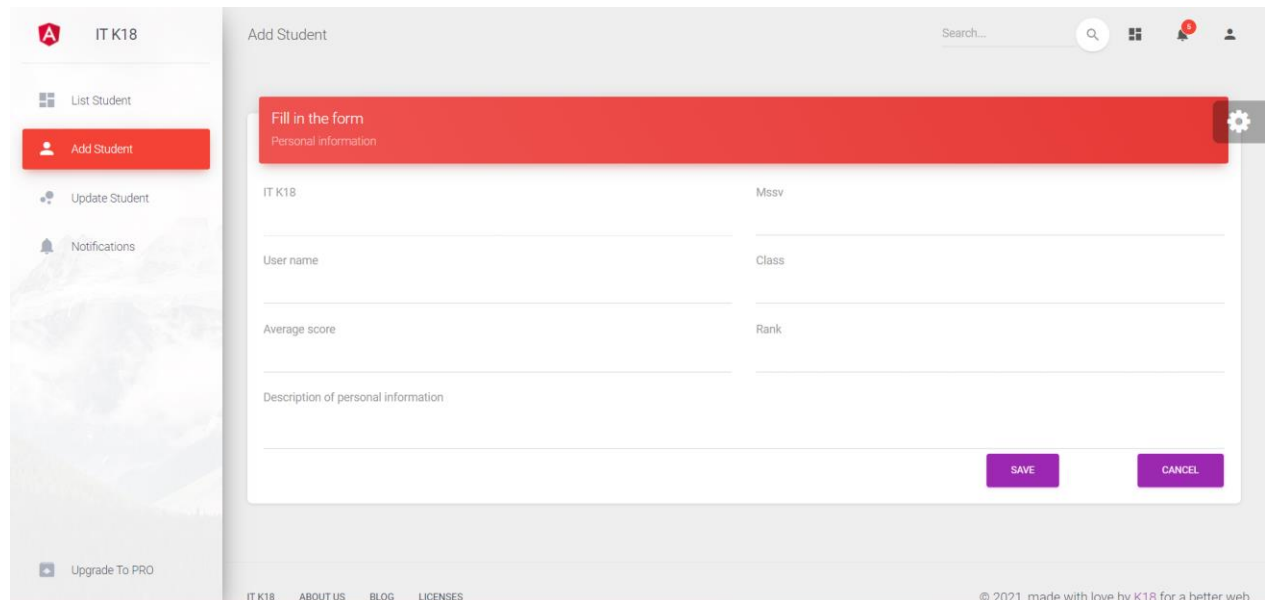
Vì xây dựng web theo mô hình tự sinh ra database. Do đó nhóm em sử dụng mysql và pull sẵn từ nhà phát hành về sử dụng.

2. Giao diện web

Giao diện danh sách sinh viên



Giao diện trang thêm sinh viên



Hình 12: Giao diện chức năng thêm sinh viên

Giao diện trang update sinh viên

Update Student

Search...

Update information
Personal information

IT K18 Mssv

User name Class

Average score Rank

Description

SAVE CANCEL

IT K18 ABOUT US BLOG LICENSES © 2021, made with love by K18 for a better web.

Hình 13: Giao diện chức năng update sinh viên

Thêm sinh viên thành công

List Student

Search...

Student Management K18

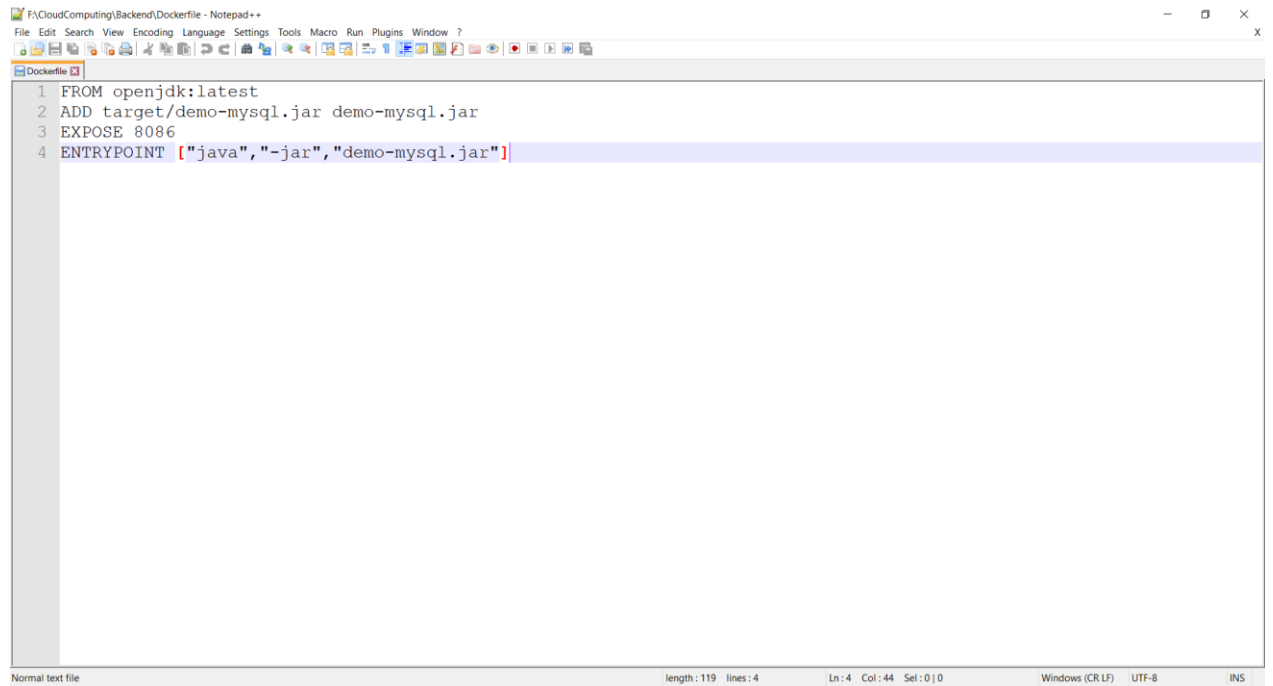
MSSV	User name	Class	Average score	Rank	
18133012	Lê Chi Hiếu	181330A	8	Giới	DETAILS/UPDATE DELETE

IT K18 ABOUT US BLOG LICENSES © 2021, made with love by K18 for a better web.

Hình 14: giao diện chức năng danh sách sinh viên sau khi thêm thành công

3. Dockerfile và docker compose của project

3.1. Dockerfile backend



```
1 FROM openjdk:latest
2 ADD target/demo-mysql.jar demo-mysql.jar
3 EXPOSE 8086
4 ENTRYPOINT ["java", "-jar", "demo-mysql.jar"]
```

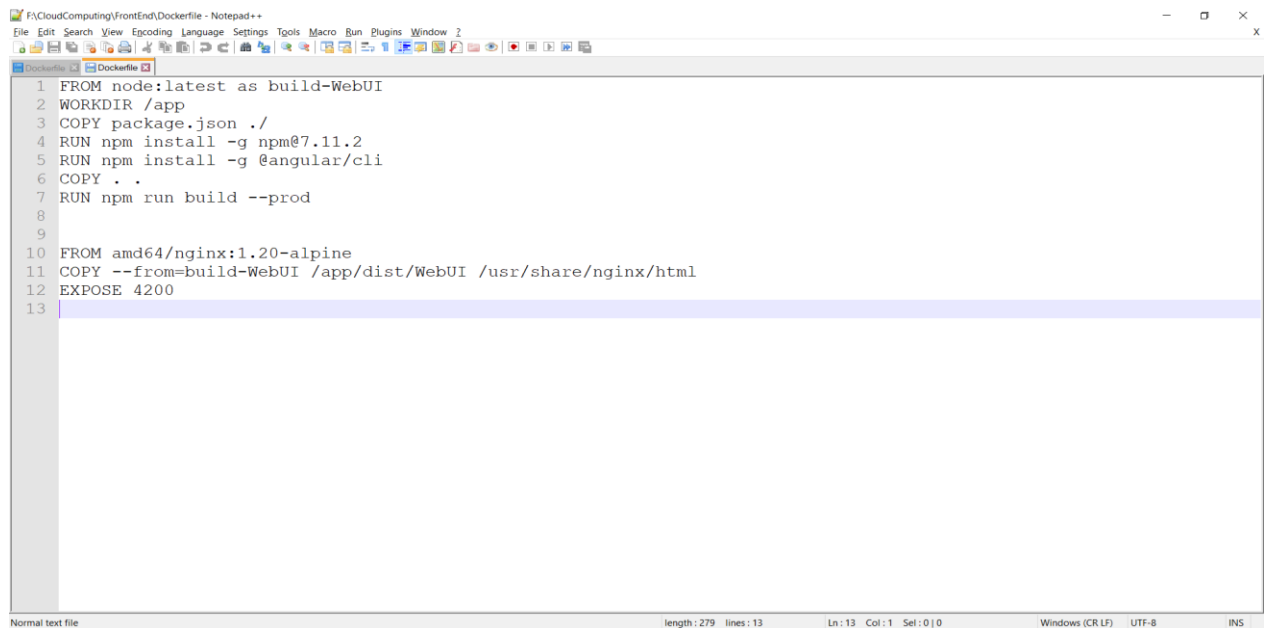
Hình 15: Docker file backend

Từ openjdk java với version mới nhất

Thêm thư mục target/demo-mysql.jar với tên là demo-mysql.jar

Expose 8086 cổng được sử dụng.

3.2. Dockerfile frontend



```
1 FROM node:latest as build-WebUI
2 WORKDIR /app
3 COPY package.json ./
4 RUN npm install -g npm@7.11.2
5 RUN npm install -g @angular/cli
6 COPY . .
7 RUN npm run build --prod
8
9
10 FROM amd64/nginx:1.20-alpine
11 COPY --from=build-WebUI /app/dist/WebUI /usr/share/nginx/html
12 EXPOSE 4200
13
```

Hình 16: Docker file frontend

Từ node version mới nhất với tên là build-WebUI.

WORKDIR /app: chỉ ra thư mục làm việc.

COPY package.json ./: copy package.json vào thư mục app.

RUN npm install -g [npm@7.11.2](#): tiến hành cài đặt npm cho angular.

RUN npm install -g @angular/cli: cài đặt gói angular/client version mới nhất.

COPY . .: Copy tất cả thư mục.

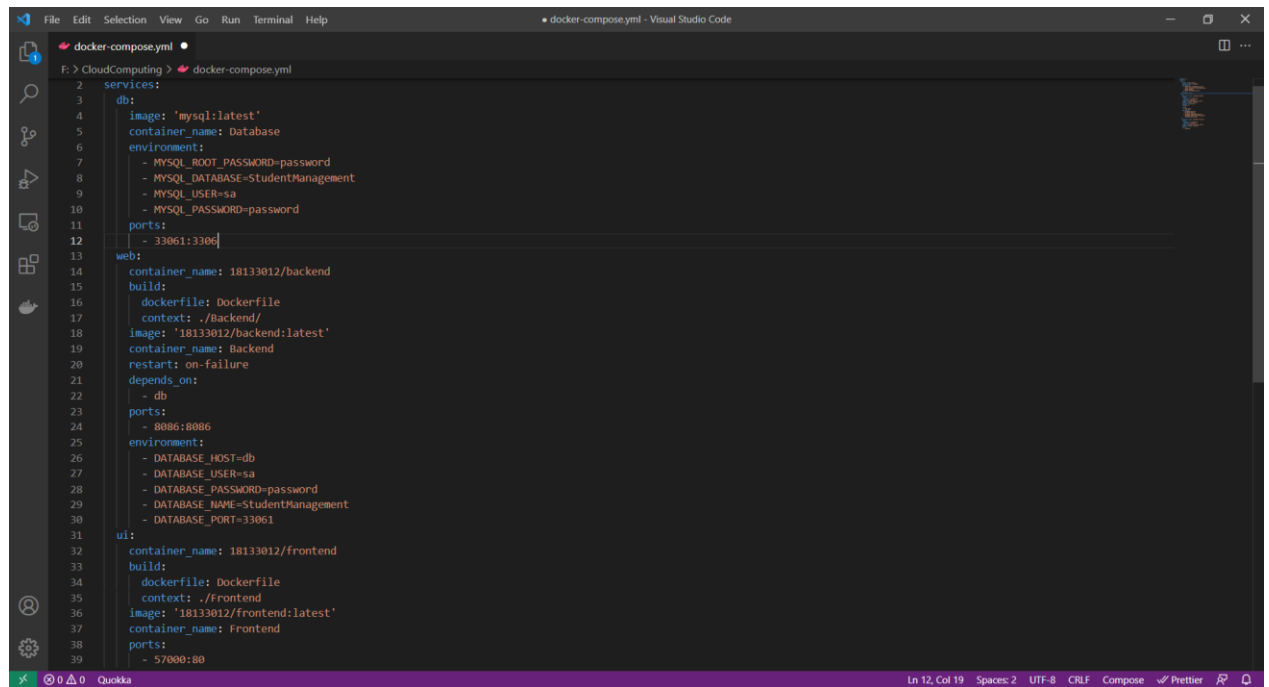
RUN npm run build --prod: Build project.

FROM amd64/nginx:1.20-alpine: Xây dựng server cho frontend.

COPY --from=build-WebUI /app/dist/WebUI /usr/share/nginx/html: Copy các file từ WebUI đến nginx.

EXPOSE 4200: Cổng được sử dụng.

3.3. Docker compose.

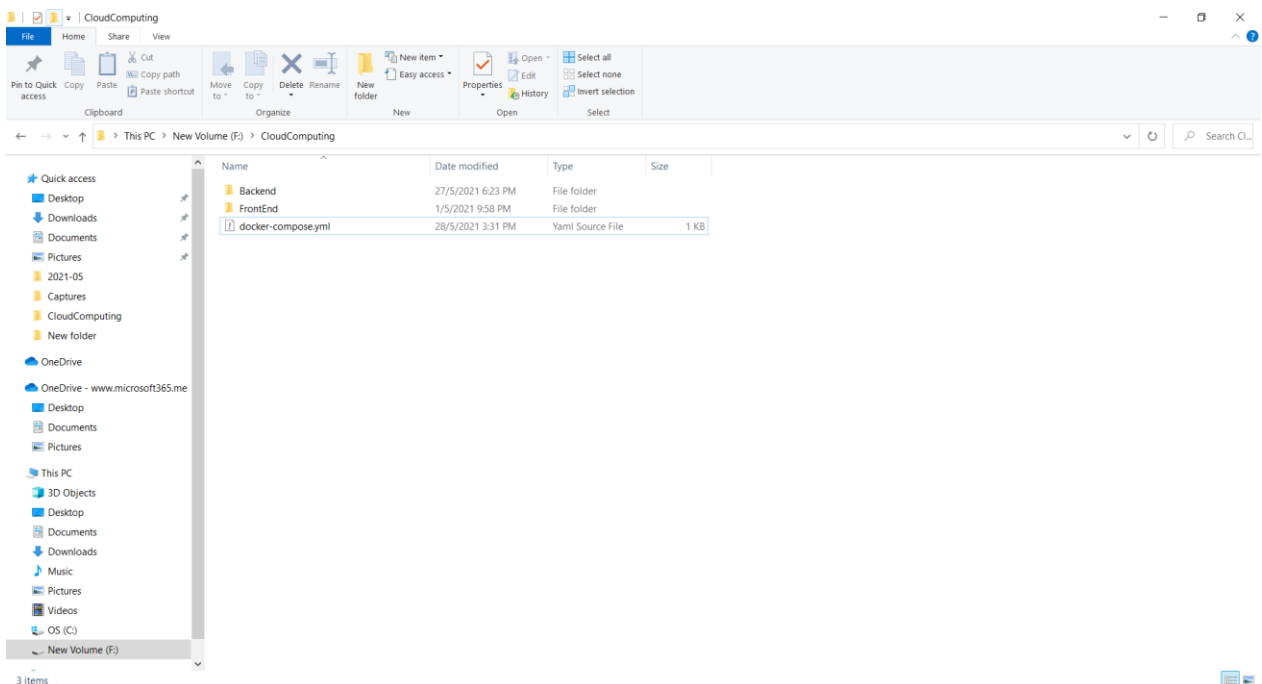


Hình 17: Docker compose

- version: '3': phiên bản docker-compose 3 đã sử dụng.
- services: thiết lập các services(containers) muốn cài đặt và chạy.
- **DB**
- image: 'mysql:latest': image mysql version mới nhất được sử dụng trong lúc tạo ra container.
- container_name: Database: tên của container trong lúc chạy.
- environment: thiết lập môi trường database với tài khoản, tên database được tạo, user sử dụng và password.
 MYSQL_ROOT_PASSWORD=password
 MYSQL_DATABASE=StudentManagement
 MYSQL_USER=sa
 MYSQL_PASSWORD=password
- ports: thiết lập port chạy trong máy host và trong container
 - 33061:3306
- **WEB**
- container_name: 18133012/backend: tên container trong lúc chạy.
- build: build từ thư mục dockerfile và đường dẫn nằm trong thư mục Backend.
 dockerfile: Dockerfile
 context: ./Backend/
- image: '18133012/backend:latest': image 18133012/backend version mới nhất được sử dụng trong lúc tạo ra container.
- container_name: Backend: tên container trong lúc chạy

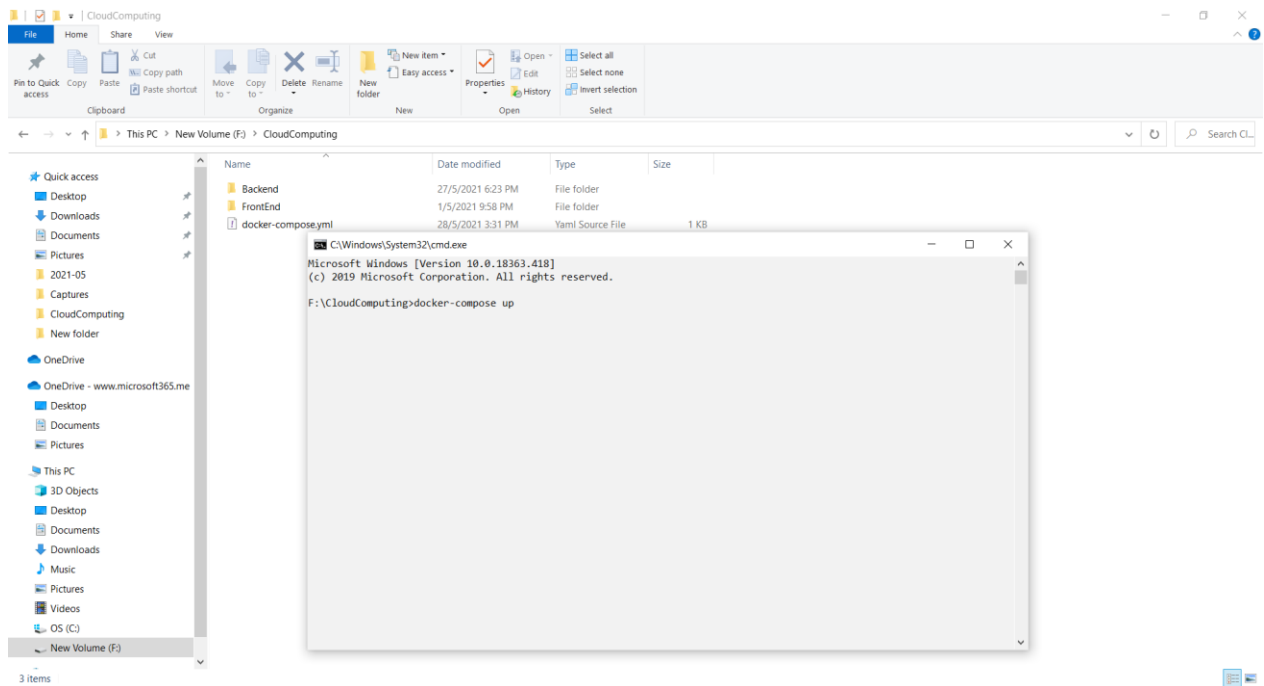
- restart: on-failure: khởi động lại container khi bị lỗi.
- depends_on: phụ thuộc vào db nghĩa là db phải được tạo trước rồi backend mới chạy.
 - db
- ports: thiết lập port chạy trong máy host và trong container.
 - 8086:8086
- environment: thiết lập môi trường database tên database được tạo, user sử dụng, password và port của mysql mặc định là 33061.
 - DATABASE_HOST=db
 - DATABASE_USER=sa
 - DATABASE_PASSWORD=password
 - DATABASE_NAME=StudentManagement
 - DATABASE_PORT=33061
- **UI**
- container_name: 18133012/frontend: tên container build
- build: build từ thư mục dockerfile và đường dẫn nằm trong thư mục Frontend.
 - dockerfile: Dockerfile
 - context: ./Frontend/
- image: '18133012/frontend:latest': image 18133012/frontend version mới nhất được sử dụng trong lúc tạo ra container.
- container_name: Frontend: tên container trong lúc chạy
- ports: thiết lập port chạy trong máy host và trong container.
 - 57000:80

4. Hướng dẫn sử dụng



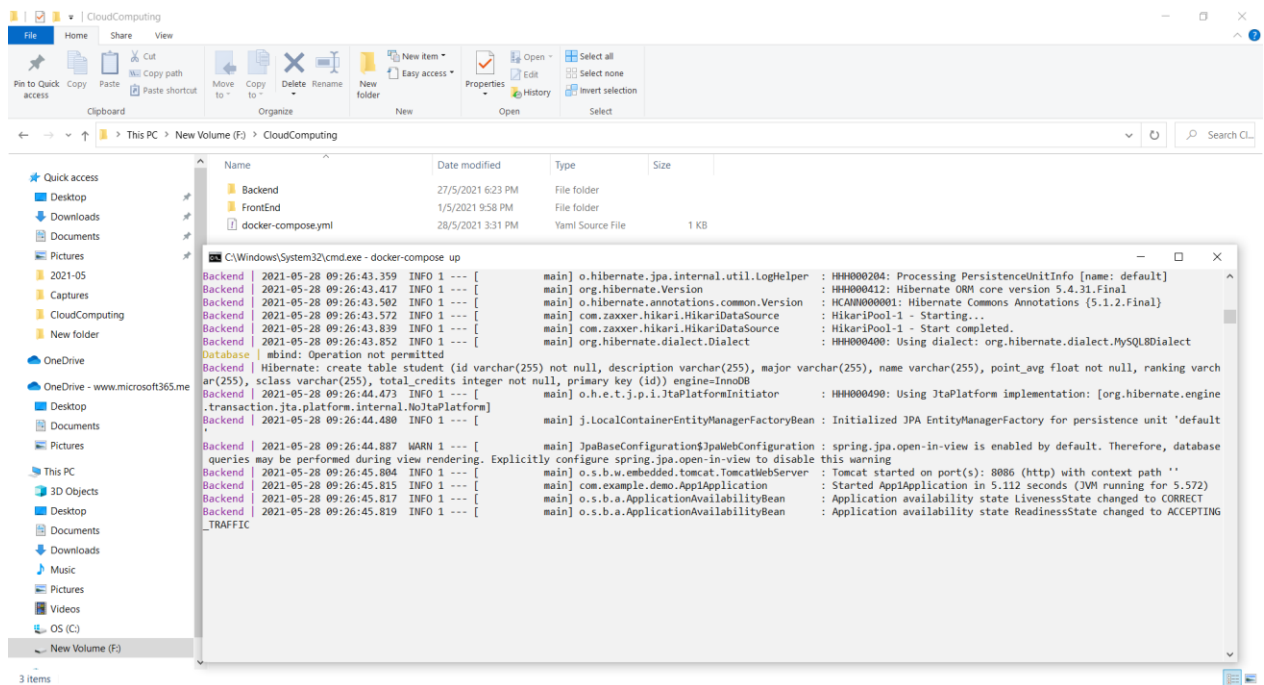
Hình 18: Hướng dẫn sử dụng chạy docker-compose.yml

Từ thư mục CloudComputing tiến hành chạy docker-compose.yml.

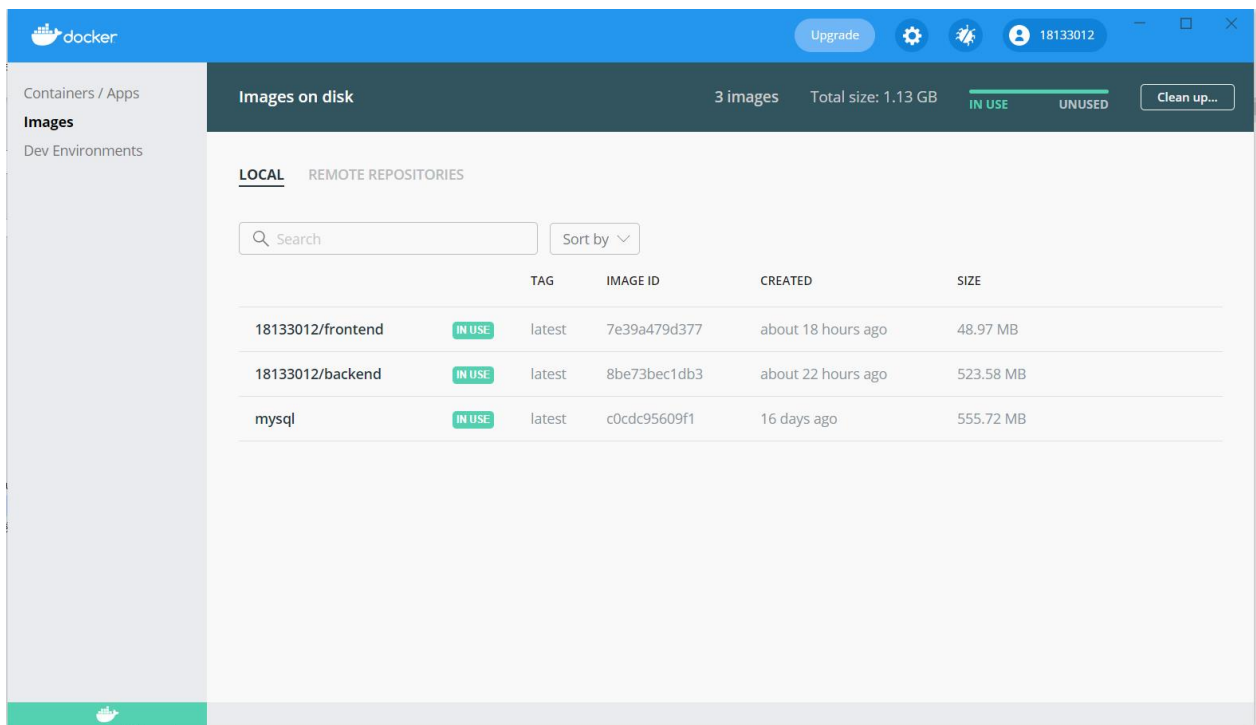


Hình 19: Chạy docker-compose.yml trên terminal

Chạy docker-compose bằng cách mở terminal lên mà trở tới thư mục CloudComputing. Sau đó chạy lệnh “docker-compose up”.

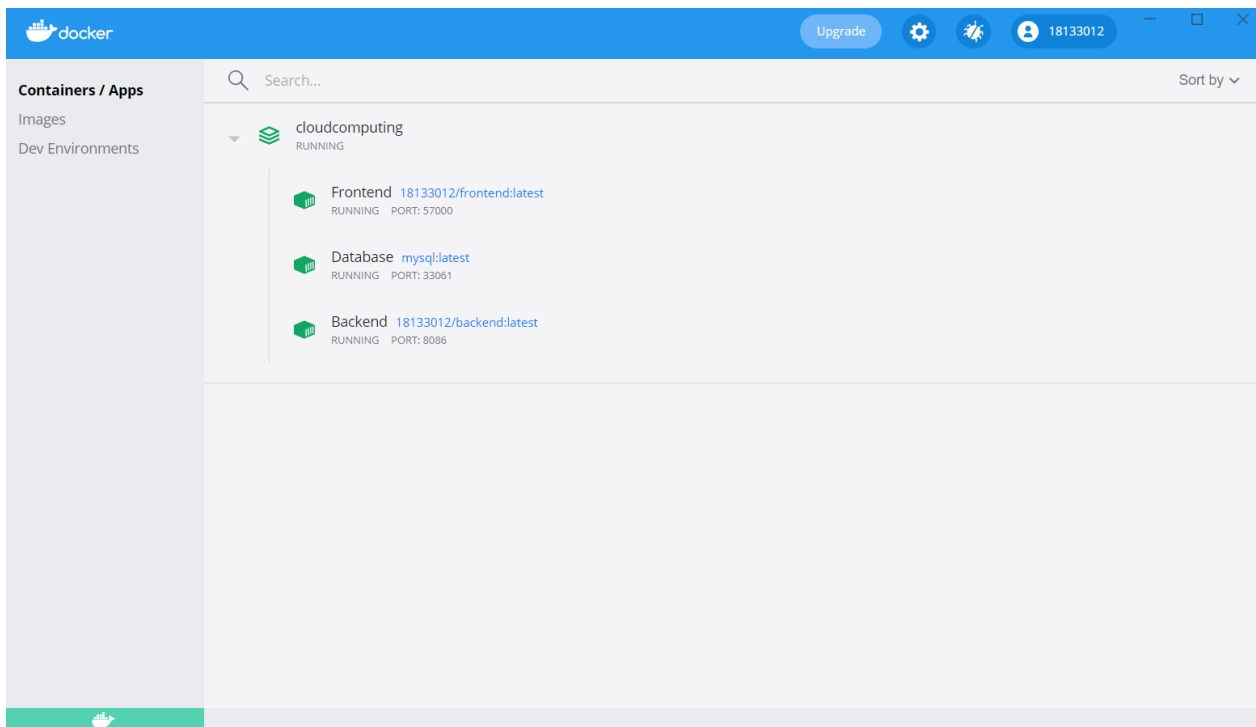


Sau khi build xong thì ta được như sau.



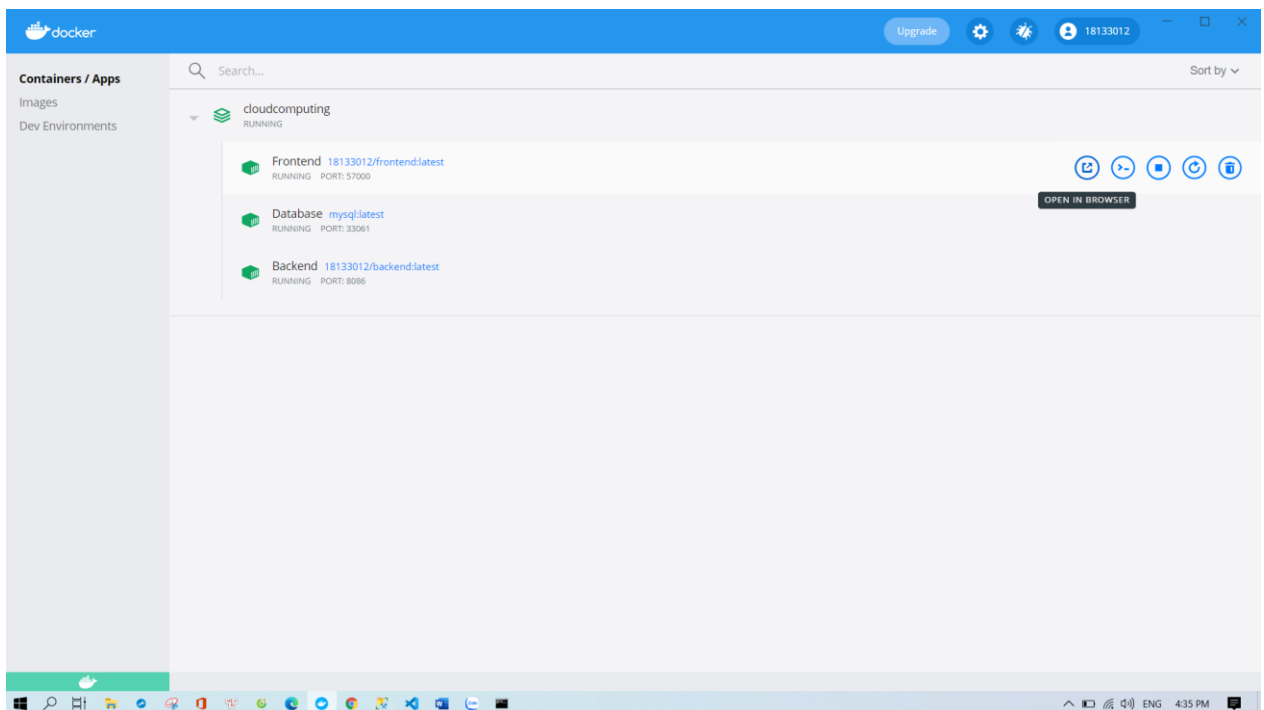
Hình 20: Các images sau khi build thành công

Thì ta thấy có các images: frontend, backend và database.

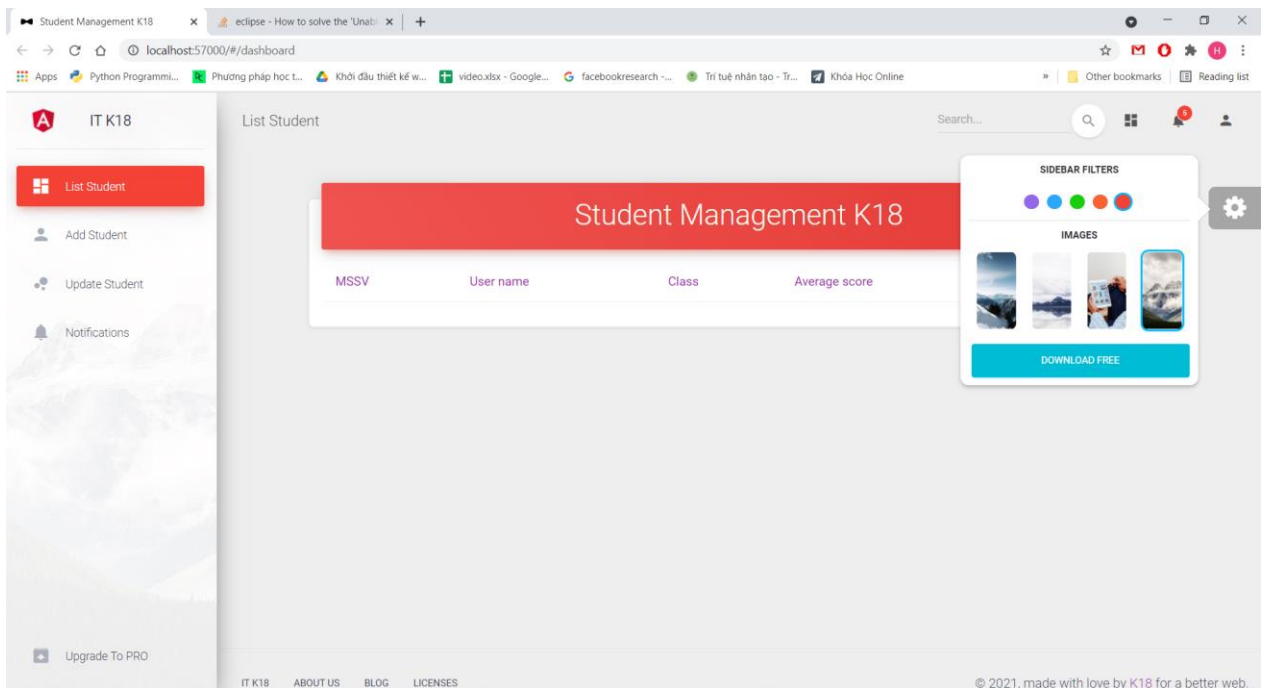


Hình 21: Các container sau khi build thành công

Đây là container được build thành công.



Để chạy lên giao diện ta chọn open in browser. Kết quả được như hình dưới.



Chương 5: Kết luận và hướng phát triển

1. Kết luận

1.1. Ưu điểm

- Ứng dụng demo được web quản lý sinh viên
- Sử dụng được dockerfile, docker-compose, container, images.
- Cho thấy được điểm mạnh của docker.

1.2. Nhược điểm

- Còn hạn chế một số lỗi như chức năng còn ít
- Vì nginx còn bị xúu lỗi trong quá trình load trang.

2. Hướng phát triển

- Xây dựng web với chức năng hoàn thiện hơn
- Phát triển cho các mô hình ứng dụng tương tự

Tài liệu tham khảo

- [1] Lệnh docker-compose tạo và chạy các dịch vụ Docker | Author: XuanThuLab-
<https://xuanthulab.net/lenh-docker-compose-tao-va-chay-cac-dich-vu-docker.html>
- [2] Docker compose là gì ?, Authour: VIBLO
<https://viblo.asia/p/docker-compose-la-gi-kien-thuc-co-ban-ve-docker-compose-1VgZv8d75Aw>
- [3] Angular Application With Bootstrap. Author: Ahmed Bouchefra-
<https://www.smashingmagazine.com/2019/02/angular-application-bootstrap/>

