Assignment 1

September 17, 2024

1 Problem 1

Repo Link: https://github.com/mgutierrezc/DS6600_lab1

Adding .gitignore lines related to the environment

```
# Environments
.env
.venv
env/
venv/
ENV/
env.bak/
venv.bak/
```

2 Problem 2

2.1 Problem 2.a.

Pipfile Content

```
[[source]]
url = "https://pypi.org/simple"
verify_ssl = true
name = "pypi"

[packages]
requests = "==2.31.0"
numpy = "==1.25.2"
pandas = "==2.0.3"
matplotlib = "==3.7.2"
jupyterlab = "==4.0.5"
ipykernel = "==6.25.1"

[dev-packages]

[requires]
python_version = "3.11"
python_full_version = "3.11.4"
```

2.2 Problem 2.b.

[1]:
```
!python -V
```

```
Python 3.11.4
```

2.3 Problem 2.c.

[2]:
```
import numpy, pandas, matplotlib, requests
```

2.4 Problem 2.d.

[3]:
```
import sklearn
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[3], line 1
----> 1 import sklearn

ModuleNotFoundError: No module named 'sklearn'
```

3 Problem 3

3.1 Problem 3.a.

- `Containers` works best as it'll allow the software to be generalized for all chapters and, by pushing it to the hub, any chapter would be able to use the portal and app plus setting up their databases.
- Neither of the other options allow the same level of generalization nor compatibility, with the exception of a virtual machine (vm onwards). However, sharing a vm is complicated given their sizes, don't have the same stability and take out a lot of resources from the computer deploying them.

3.2 Problem 3.b.

- All options are overkill but the virtual and global environment. However, using a global environment for any task is a bad practice. Installing any packages here always carries the risk of altering some component in your installation of python, which could end up in a situation where you are no longer able to run anything globally unless reinstalling python. Hence, for this and similar situations, it's best to work on a `virtual environment`.
- The other options are overkill because you've been asked for the report only, not a system capable of deploying the tools used to create it.

3.3 Problem 3.c.

- A `virtual machine` is the best solution as it can allow you to debug the code you're working on in the right OS, and specially if you are working by yourself. It'd be better if instead you

deploy your project in a cloud server though as it allows for more flexibility and consumes less local resources (only resources needed are the ones to code).
- A container would be overkill as it might only be necessary only for the final deployment, as it's main advantage is the deployment of fully developed services.
- A virtual/global environment would be insufficient given that they lack of control regarding the OS used for their deployment.

3.4 Problem 3.d.

- A `virtual environment` should be sufficient as you can create one with any python version you want.
- The global environment can work in this scenario too as you can have multiple python versions installed on your computer simultaneously and use the `py` manager to switch to the one you need at any point. You can do this and create a virtual environment after choosing the python version you wanna work with.
- The other options are an overkill as you'd be consuming much more resources to deploy Python 4 when you have easier/less intensive options such as the ones described above.

4 Problem 4

4.1 Problem 4.a.

```
FROM ubuntu:latest

# update package list
RUN apt-get update && apt-get install -y python3

# Set the default command to run Python 3
CMD ["python3"]
```

4.2 Problem 4.b.

```
PS D:\Accesos directos\Trabajo\UVA\Classes\DEng I\Repos\DS6600_lab1> docker build . -t assignme
[+] Building 19.9s (7/7) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 194B
 => [internal] load metadata for docker.io/library/ubuntu:latest
 => [auth] library/ubuntu:pull token for registry-1.docker.io
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/2] FROM docker.io/library/ubuntu:latest@sha256:dfc10878be8d8fc9c61cbff33166cb1d1fe44391
 => => resolve docker.io/library/ubuntu:latest@sha256:dfc10878be8d8fc9c61cbff33166cb1d1fe44391
 => => sha256:dafa2b0c44d2cfb0be6721f079092ddf15dc8bc537fb07fe7c3264c15cb2e8e6 29.75MB / 29.75
 => => sha256:dfc10878be8d8fc9c61cbff33166cb1d1fe44391539243703c72766894fa834a 1.34kB / 1.34kB
 => => sha256:77d57fd89366f7d16615794a5b53e124d742404e20f035c22032233f1826bd6a 424B / 424B
 => => sha256:b1e9cef3f2977f8bdd19eb9ae04f83b315f80fe4f5c5651fedf41482c12432f7 2.30kB / 2.30kB
 => => extracting sha256:dafa2b0c44d2cfb0be6721f079092ddf15dc8bc537fb07fe7c3264c15cb2e8e6
 => [2/2] RUN apt-get update && apt-get install -y python3
```

```
 => exporting to image
 => => exporting layers
 => => writing image sha256:9478eacbf6c7e59d8693385c0954ae38e5535174d27dfab38b1691e120165d53
 => => naming to docker.io/library/assignment_1_p4

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview
```

4.3 Problem 4.c.

```
PS D:\Accesos directos\Trabajo\UVA\Classes\DEng I\Repos\DS6600_lab1> docker run -it assignment_
Python 3.12.3 (main, Sep 11 2024, 14:17:37) [GCC 13.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

5 Problem 5

Docker Hub link: https://hub.docker.com/repository/docker/mgutierrezc442/assg1-p5/general

6 Problem 6

It is written in the Book of Tyr:

```
After the Creation, the cruel god Moloch rebelled
against the authority of Marduk the Creator.
Moloch stole from Marduk the most powerful of all
the artifacts of the gods, the Amulet of Yendor,
and he hid it in the dark cavities of Gehennom, the
Under World, where he now lurks, and bides his time.
```

Your god Tyr seeks to possess the Amulet, and with it to gain deserved ascendance over the other gods.

You, a newly trained Stripling, have been heralded from birth as the instrument of Tyr. You are destined to recover the Amulet for your deity, or die in the attempt. Your hour of destiny has come. For the sake of us all: Go bravely with Tyr!