Docker documentation

Created a docker container image named *dockerfile* to include the following steps:

- Step 1 - Use an official Python runtime as a parent image. (used `python:3.8-slim`)
- Step 2 - Set the working directory in the container. Used WORKDIR /app.
- Step 3 - Copy the application files in the container. Used the command (COPY . .)
- Step 4 - Install system dependencies and ODBC driver.
- Step 5 - Install pip and setuptools.
- Step 6 - Install Python packages specified in requirements.txt.
- Step 7 - Expose port (EXPOSE 5000)
- Step 8 - Define Startup Command (CMD ["python", "app.py"])

Next, build the docker container image with the command:

```
>docker build -t dockerfile .
```

Next, run the docker container using the command that includes the exposed port as in the docker image:

```
>docker run -p 5000:5000 dockerfile
```

Next, Open a web browser and go to http://127.0.0.1:5000 to interact with the application within the Docker container to confirm the application works.

Next, Tag and push the docker container image to docker hub. First login to docker using the command in the command line: docker tag dockerfile whule/dockerfile (note that dockerfile is the name of the docker container image).

Then push the container image to docker hub with this command: docker push whule/dockerfile (Note that whule is my docker hub userid and dockerfile is the docker container image name)

To clean up the images and container, used the docker ps -a command to list all containers, including stopped ones. Removed any unnecessary containers with docker rm <container-id> to free up resources.

To remove Images: Listed all images using docker images -a and removed any unneeded images with docker rmi <image-id> to reclaim disk space.