Python Developer Technical Challenge

Description: This code test consists of the implementation of a simple Python Flask API using a backend Sqlite database. We would like a service that is:

- **Unit/functionally tested**
- **Stable**
- **Secure**
- **Maintainable**.

Technical Spec:

Given the test dataset below, please implement:

```
"make","model","year","chassis_no" ,"id" ,"last_updated",”price”
"Nissan","Micra",2004,"12345A",1,"2017-02-01 00:00:00", 500.0
"Nissan","Micra",2004,"12425A",1,"2017-03-01 00:00:00", 400.0
"Ford","Fiesta",2002,"12345B",2,"2017-03-01 00:00:00", 300.0
"Audi","A3",,"12345C",3,"2017-04-01 00:00:00",
"Nissan","Micra",2004,"12345D",4,"2017-05-01 00:00:00", 200.0
"Peugeot" ,"308",1998,"12345E",5,"2017-06-01 00:00:00", 100.0
```

1. Use Python 3.x
2. Endpoint to retrieve a record by id. Do not return the ‘chassis_no` in the response.
3. Endpoint to return the average price by *make* or *model* or *year,* or a combination of the above.
5. Please include the DDL and DML to create the database and table and load from csv above.

Extra:

- You implement a swagger endpoint to describe and document the API.
- You are able to run the API as a container using Docker.
- You are able to run the API and the database in separate containers using docker-compose.

Evaluation:

1. Your code challenge will be evaluated if committed to GitHub. Please add relevant messages to your commit through-out the development.
2. Functionality and overall architecture will be assessed.
3. Make sure you add a **README.md** file with the appropriate documentation including
    a. *Description of the API*

b. *How to run the app*
c. *Curl* requests example and responses.
d. *Simple diagram of the architecture.*

Resources:

1. Flask *http://flask.pocoo.org/*
2. SQLAlchemy *http://flask-sqlalchemy.pocoo.org/*
3. Docker *https://www.docker.com/*
4. 12 factor apps principles *https://12factor.net/*