Using Python **Object Oriented Programming** complete the following exercises

First part:

Download the information for the following products:

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 500, 501]

Using the API

https://reqres.in/api/products/10

ID of products required.

Note = Product 500/501 doesn't have data.

If the product id doesn't exists/no data, add the product into the data frame only with the ID, and the rest of the information should be null

Columns needed

[#id, name, year, color, pantone_value#,

11, "turquoise", 2010, "45B5AA", "15-5519",

501, nan, nan, nan, nan, nan]

The data frame must have 13 rows

After the information is downloaded into a dataframe
Modify the year column. **Modify the year 2010 to 2099 in the data frame**

Calculate the median year of the products and print it out using a print statement to look like below output

Sample Output:

"Median year of product is 2012"

Second part:

Create a GET custom web API named: get_product_info(product_id)
using any web framework(flask/django/another) that you feel comfortable on python
that pulls information for the products from the API (https://reqres.in/api/products/?)

Business logic:

If the product ID is less than 5, return the original json object for that product (sample return object below)

Note:

If the user provides a valid json object on the get request containing the key {"insert_db" =True}
Return an additional key called on the product result with the name "Uploaded_db" and the current DateTime with format yyyy-mm-dd hh:mm:ss

{

"data": {

"id": 1,

"name": "cerulean",

"year": 2000,

"color": "#98B2D1",

"pantone_value": "15-4020"

},

"support": {

"url": "https://reqres.in/#support-heading",

"text": "To keep ReqRes free, contributions towards server costs are appreciated!"

},

"*Uploaded_db*": "2021-07-25 10:40:10"

}

If the productID is greater or equal than 5, return the following json object and add additional key call "EVALUATION" with the value of "TESTING" (sample return object below)

```
{
        "data": {
        "id": 1,
        "name": "cerulean",
        "year": 2000,
        "color": "#98B2D1",
        "pantone_value": "15-4020",
        "EVALUATION": "TESTING"
        }
}
```


If the product doesn't have data=

Return a json dictionary with the id of the product and the key "Note"

Example:


```
{
        "id": 501,
        "note": "No data available"
}
```


If the product id is not numeric, return an error 405 on the API endpoint

Create a docker-compose file to run the above web project locally running on port 3001

Extra points:

On the same project,
Create an API endpoint call get_download_product(product_id) where you will need to return an excel file with the following columns + an extra column call download_time with the current time

['id', 'name', 'year', 'color', 'download_time'

1, "cerulean", 2000, "#98B2D1", '2021-07-25 10:40:10"]

The name of the file to be downloaded should be "Product_Download_{product_id}.xlsx"

Expected deliveries:

- Requirements to run the project
- Python code
- Docker-compose file
- Git commands to upload the information into a dev branch
- If there was a question that was not completed/fully understand, please provide the details