Assignment # 5

Due Date Wednesday, April 15, 2021, @11: 59 PM

In this Assignment, you will modify **the product management app** app's backend that is currently using file-based repository to read and write data with mongo database.

Preparation

1. Sync cmps350-lab repo to get the Assignment files
2. Open the **assignment5/ProductManagementApp** project
3. The project contains the complete implementation of assignment 4
4. Run **npm install** to install all the necessary packages.

Implementing the Databas

In this part you should implement the entity schemas and the database repository for the **ProductManagementApp**. The database repository should implement the same functionality as the file-based repository provided in the base solution.
Note : You can not use any of the array function such as **filter**, **map** , **reduce** etc… to filter or aggregate values. You should use the **mongoose model methods** to get, delete, update , aggregate data.

Method	Description
getProducts()	Return all products from the database
getProduct(pid)	Return a single product that has the same product id(pid)
addProduct(newProdcut)	Adds new product to the database
updateProdcut(updatedProduct)	Updates existing product
deleteProduct(pid)	Deletes specific product from the database
deleteAllProducts()	Remove all the products saved inside the database

getStatistics()
 	
[you must **use aggregate function** to implement this. Using array functions such as **reduce, map** will not be accepted]
	Returns an object containing the total number of products in the supermarket and their total price.
	Example object should look like
	{
	totalNumberOfProducts : 5,
	totalPrice : 500
	}

getTopExpensiveProducts(**limit**) [you must **use aggregate function** to implement this. Using array functions such as **reduce, map** will not be accepted]	Returns the top **X** expensive products in the database. For instance if you call the function getTopExpensiveProducts(3) then it should return the top three expensive products in the database products collection.

- Test your API's using **Postman and the Client side application**

After you complete the Assignment, fill in the ***TestingDoc-Grading-Sheet.docx*** and save it inside the ***Assignment5*** folder. Push your work to the Github repository.