**.Net Web Api Evaluation**  
  
**Instructions: Follow the next steps where you are going to create a webapi with some features, when you are finish create a public repo on Github,Gitlab etc. And push your changes/project to that repo also share the link to the repo on you Academy folder by uploading the txt with the url inside of it**  
  
Create a .Net Core (.Net) web api project

We will create the following endpoints

localhost/user

localhost/inventory

User Object Properties:  
User {UserId, Name, UserType(Enum), Email, Phone}

The endpoint user will have the following methods  
METHOD/URL

POST:user/add , GET:user/get, UPDATE: user /update , DELETE: user /delete

The user endpoint will create, get, update and delete a user in an in memory Database (static object/class).

add: It will receive an object in json format and will save a user in the database and return a json of the user created with ID generated by the database

get: this endpoint will receive a json type object, or a query string or a url variable that specifies the user to get and this will return the database user in json format

update: will receive a user object in json format which will have the updated parameters and the object will be updated in the database, this endpoint will return the updated user in json format

delete: will receive an object, query string or variable in url that corresponds to the ID of the user to delete and this will be deleted from the database and the information of the deleted user will be returned in json format

The inventory endpoint will be an endpoint where you can add, update, get, getAll, getByUser and delete an item(s) list from the user's inventory in a static list, the Inventory endpoint methods will share the same logic as the user endpoint.

Inventory Item Object Properties  
Item {ItemId, Name, Description, Quantity, UserId(The user that this item belongs to)}  
  
getAll: get all will return all the items from all the users from the inventory list  
getByUser: this endpoint will received a json type object, query string or url variable that specifies the UserId to retrieve its inventory and will return the list of items that belong to this user on json format  
  
**Extra: Add a new functionality to the inventory endpoint and leave the description of the functionality as a comment on the controller**