LAB 1

Create an **inventory management CLI tool** that allows users to manage stock items through terminal commands.

1 - Users can add an item to inventory

Example:

node index.js add "Laptop"

Notes:

Each item should have an **incremental id**

Items are added with a default:

- quantity = 1
- category = "General"

2 - Users can destock an item using its id

Destocking means **reducing the quantity** of an item when stock is removed (sold, damaged, etc.).

Example:

node index.js destock [id] [quantity]

node index.js destock 232 2

3 - Users can restock an item

Restocking means **increasing the quantity** of an item.

Example:

node index.js restock [id] [quantity]

node index.js restock 232 3

4 - Users can edit an item name using its id

Example:

node index.js edit 232 “New Name”

5 - Users can remove an item completely

Example:

node index.js delete 232

6 - Users can list all the items

Example:

node index.js list

Note:

Each item should have a status

Quantity > 2 ===> status = “available”

2 > Quantity > 0 ===> status = “low stock”

Quantity = 0 ===> status = “out of stock”

Do not print the raw array

Items should be displayed in a **clear, readable format** in the terminal (aligned text, multiple lines per item, or a table-like view)

7 - Summary command

Add a command that shows a **summary of the inventory**.

Example:

node index.js summary

The summary should include:

Total number of items

Total quantity of all items

Number of items per status (available - low stock - out of stock)

BONUS

You can use a NPM package to parse the cli argv

1- add the items with quantity and category options

Keep the defaults if options not provided

Example:

node index.js add "Laptop" -q 2 -c “electronics”

node index.js add "Laptop" --quantity 2 --category “electronics”

2- edit the items name and category

node index.js edit 123 -n “new name” -c “electronics”

3- Unique Item Name Validation

Each inventory item must have a unique name.

Rules:

Item names are case-insensitive

"Laptop" and "laptop" are considered the same item

Users cannot add or edit an item with a name that already exists

4- List items by status and category

Allow filtering with status or category or both

Example: node index.js list -s “available” -c “electronics”

Notes:

- Please make sure that you separate your logic in functions
- Make the names of your functions and variables are expressive of what they are
- Start by defining each functionality and the steps you should make to achieve it
- Please make sure not to pollute the global scope[refrain from using global variables]
- Before writing a function, ask yourself, is there a function in javascript that can do that, if not, write your own.
- You initialize your app with npm init to be able to use npm and download packages
- Take a minute or two to read the package.json file

Tips:

- If you can't do something, search, then ask for help, this is not an exam.
- Please please please, don't copy someone else's code.
- Make sure to take a few minutes to read about the commander library that you'll be using.
- Start with implementing the required then move to the bonus

Useful Links:

- npm | build amazing things
- Nodejs Docs
- Fs module docs fs (you will need to use the sync version of the functions)[ends with the word Sync]
- Cli helper library name is commander, search for it at npm.