BACKEND CHALLENGE

Your code will be evaluated on cleanliness, organization, completeness, and extensibility, among other factors. Good luck and please document how much total time you spent on the challenge when you are ready to submit!

I. SUMMARY:

A client is building an e-commerce mobile application for their line of coffee machines and custom coffee pods; they are looking to have two screens: one screen to display coffee machines and one screen to display coffee pods. On the coffee machines screen, the user may filter by product type and water line. On the coffee pods screen, the user may filter by product type, coffee flavor, and pack size. Your task is to simulate this environment and build an API to return the data for these two screens.

II. PRODUCT SCHEMA:

Coffee machines

· Product type – *product_type*
o COFFEE_MACHINE_LARGE
o COFFEE_MACHINE_SMALL
o ESPRESSO_MACHINE

· Water line – *water_line_compatible*
o true
o false

Coffee pods

· Product type – *product_type*
o COFFEE_POD_LARGE
o COFFEE_POD_SMALL
o ESPRESSO_POD

·Coffee flavor – *coffee_flavor*
o COFFEE_FLAVOR_VANILLA
o COFFEE_FLAVOR_CARAMEL
o COFFEE_FLAVOR_PSL
o COFFEE_FLAVOR_MOCHA
o COFFEE_FLAVOR_HAZELNUT

· Pack size – *pack_size*
o 1 dozen (12)
o 3 dozen (36)
o 5 dozen (60)
o 7 dozen (84)

III. SKU LIST
· CM001 – small machine, base model
· CM002 – small machine, premium model
· CM003 – small machine, deluxe model, water line compatible
· CM101 – large machine, base model
· CM102 – large machine, premium model, water line compatible
· CM103 – large machine, deluxe model, water line compatible
· EM001 – espresso machine, base model
· EM002 – espresso machine, premium model
· EM003 – espresso machine, deluxe model, water line compatible
· CP001 – small coffee pod, 1 dozen, vanilla
· CP003 – small coffee pod, 3 dozen, vanilla
· CP011 – small coffee pod, 1 dozen, caramel
· CP013 – small coffee pod, 3 dozen, caramel
· CP021 – small coffee pod, 1 dozen, psl
· CP023 – small coffee pod, 3 dozen, psl
· CP031 – small coffee pod, 1 dozen, mocha
· CP033 – small coffee pod, 3 dozen, mocha
· CP041 – small coffee pod, 1 dozen, hazelnut
· CP043 – small coffee pod, 3 dozen, hazelnut
· CP101 – large coffee pod, 1 dozen, vanilla

· CP103 – large coffee pod, 3 dozen, vanilla
· CP111 – large coffee pod, 1 dozen, caramel
· CP113 – large coffee pod, 3 dozen, caramel
· CP121 – large coffee pod, 1 dozen, psl
· CP123 – large coffee pod, 3 dozen, psl
· CP131 – large coffee pod, 1 dozen, mocha
· CP133 – large coffee pod, 3 dozen, mocha
· CP141 – large coffee pod, 1 dozen, hazelnut
· CP143 – large coffee pod, 3 dozen, hazelnut
· EP003 – espresso pod, 3 dozen, vanilla
· EP005 – espresso pod, 5 dozen, vanilla
· EP007 – espresso pod, 7 dozen, vanilla
· EP013 – espresso pod, 3 dozen, caramel
· EP015 – espresso pod, 5 dozen, caramel
· EP017 – espresso pod, 7 dozen, caramel

IV. USE CASES

The first task is to determine what the return array should look like for the certain cross-sell scenarios (see reference examples).

· All large machines
o CM101
o CM102
o CM103

· All large pods
o CP101
o CP103
o CP111
o CP113
o CP121
o CP123
o CP131
o CP133
o CP141
o CP143

· All espresso vanilla pods
o EP003
o EP005
o EP007

· All espresso machines
o ?

· All small pods
o ?

· All pods sold in 7 dozen packs
o ?

V. FUNCTION IMPLEMENTATION

Build a mongo DB and REST API with endpoints that return the JSON data for the coffee machines and coffee pods screens. These endpoints should be filterable by product type, water line, coffee flavor, and pack size, depending on whether the product is a coffee machine or coffee pod. For example, when filtering for large coffee machines, the endpoint should return an array with CM101, CM102, and CM103 as JSON objects.