

Project 3

► 2023

100 Possible Points

Offline Score:
N/A

 Add Comment

▼ Details

Project 3: The Mongo Mash

You may work on and submit this project with one other person.

Overview

In this project, you will translate the Project 1 UML solution into MongoDB collections, using JSON schema documents that I will provide. You will write and submit JSON documents to represent several troop, scout, adult, and cookie documents. You will then write a short Python program using the PyMongo library, allowing a user to search the various objects in the database.

Setting up the Collections

Use Compass to create a new database, "project3". Create collections for **troops**, **adults**, and **cookietypes**.

Copy and paste the contents of the **schema_troops.json** file from Canvas (in the Projects module) into the Validation tab of the troops collection in Compass. Do the same for schema_adults.json (adults collection) and schema_cookietypes.json (cookietypes collection).

You will want to review the schemas for each collection before moving on, and will refer to them when doing the next task.

Creating validated documents

Begin by creating JSON documents for the following objects. The documents must match the schemas from the previous step.

You can do this by hand, or I will allow and *encourage* you to use a Large Language Model tool like ChatGPT to generate the objects. To use an LLM in this way, you should write a prompt explaining that you are providing a MongoDB \$jsonSchema and then paste one of the schemas into the prompt. Then



(<https://csulb.instructure.com/courses/31851/modules/items/1737708>)

(<https://csulb.instructure.com/courses/31851/modules/items/1737708>)

English sentence; like, "The Troop is numbered 1234, founded on 12-1-2023, at the Long Beach community. There are no scouts or adults or allotments in the troop."

Check the output of the AI, and use more prompts to guide it to correct any mistakes. I recommend first generating objects that have no children (like the Troop example just given), and then using further prompts to add those children. (For example, "Add a scout named X to the troop, with these details.")

Once an object is complete, use Compass to insert it into the correct collection. Compass will reject the insertion if the document does not match the schema, and you will need to work with the AI to figure out why and correct the issue. One **common** issue with the AI is that it might not match up foreign keys with an actual existing object; for example, it might say that the adult "Neal Terrell" has an adultid of 5 when the adult is referenced by a certain Scout, and later might give "Neal Terrell" a different ID when generating the actual Adult object.

The objects you need to create and insert:

- These cookie types, all with a price of \$6: Thin Mints; Samoas; Tagalongs; Trefoils; Toffeetastic.
 - Thin Mints: ingredients of sugar, cocoa, and mint oil.
 - Samoas: ingredients of cocoa, coconut extract, sugar, and butter.
 - Tagalongs: ingredients of cocoa, sugar, peanut oil, peanut butter
- Troop 1234, founded 2022-09-01, from Long Beach.
 - Two scouts:
 - Naomi Nagata, born 2017-03-03, grade K.
 - Related to the adult James Holden, born 1992-06-04; role of "Co-leader" in Troop 1234.
 - Was allotted 36 boxes of Thin Mints, 36 boxes of Samoas, and 24 boxes of Tagalongs on 2023-01-19.
 - Made a payment for \$540 on 2023-03-01
 - Made a payment for \$24 on 2023-03-15
 - Grace Hopper, born 2017-01-30, grade K.
 - Related to Ada Lovelace, born 1990-10-10; role of "Treasurer" *and* "Cookie Chair" in Troop 1234
 - Was allotted 30 boxes of Thin Mints, 20 boxes of Samoas, and 10 boxes of Tagalongs on 2023-01-19.
 - Made a payment for \$360 on 2023-02-01
 - One troop allotment:
 - On 2023-01-18, 10 cases of Thin Mints, 10 cases of Samoas, 10 cases of Tagalongs were allotted.
- Troop 6789, founded 2023-01-01, from Cerritos.
 - Two scouts:
 - Barbara Liskov, born 2014-07-01, grade 3



- Was allotted 120 boxes of Thin Mints, 36 boxes of Samoas, and 60 boxes of Tagalongs on 2023-01-20.
- Was allotted 36 boxes of Trefoils, and 7 boxes of Toffeetastic on 2023-02-17.
- Made a payment for \$1296 on 2023-01-31
- Made a payment for \$198 on 2023-02-21
- Mary Sparck Jones, born 2013-12-30, grade 3.
 - Related to the adult Mary Jackson, born 1953-01-01; not a volunteer in a troop.
 - Was allotted 120 boxes of Thin Mints, 24 boxes of Samoas, and 60 boxes of Tagalongs on 2023-01-20.
 - Was allotted 60 boxes of Thin Mints, and 5 boxes of Toffeetastic on 2023-02-18.
 - Made a payment for \$1614 on 2023-03-15
- Two troop allotments:
 - On 2023-01-18, 20 cases of Thin Mints, 5 cases of Samoas, 10 cases of Tagalongs were allotted.
 - On 2023-02-15, 5 cases of Thin Mints, 3 cases of Trefoils, and 1 case of Toffeetastic were allotted.

Python application

You must now write a Python application using PyMongo to interact with this Girl Scouts database.

Your application should begin with a Main Menu of these options:

1. Troop Lookup

- Input the number of a troop. Retrieve that troop from the database, and print a "summary" view of the troop: its number, founding date, and community; the names of each scout; and the names of the adult volunteers.

2. Scout Lookup

1. Input the first and last name of a scout. Retrieve that scout from the database, and print a detailed view: the full name, birthday, and grade level; the names of the scout's adults; and the dates of each allotment, along with the type and count of each cookie received.

Since scouts are embedded in troops, you will actually have to select a troop object. But those can be quite large. For full points, you must **unwind** the scouts array and then match the **one** unwound document that has the requested name. (You may assume all scout names are unique.)

3. Sales Report

- Input the number of a troop. Print out a sales summary for that troop: a list of each scout, with the total value of all allotments for that scout.



more difficult than will be on the final exam. The most straightforward approach only using things that **we** have studied will be to:

- A. Run a pipeline on troops. Match only the requested troop.
- B. Unwind the scouts, for 1 document per scout.
- C. Unwind the allotments, for 1 document per allotment per scout.
- D. Unwind the cookies of the allotment, for 1 document per cookietype per allotment per scout.
- E. Join to the cookietypes collection, as a field called "cookie".
- F. Unwind the cookie so it becomes an object, not an array of 1 object.
- G. Project a **totalvalue** field onto the document, by multiplying the **amount** of the allotment by the **price** of the cookie.
- H. Group on the firstname and lastname of the scouts, and compute the sum of the totalvalue fields of the grouped documents.
- I. The resulting documents have a firstname, lastname, and total value of allotments.

Deliverables

You must deliver to me a printout containing the following documents:

- Title page (name of course and section number, title of assignment, team member names, **due** date)
- JSON documents for the objects you were required to create.
- The entire code of the Python application you were required to write. I should be able to run your program after inserting the documents you submitted.

