

CSE 4701, Fall 2023 Project 2

Part I: Due Nov 15, 2023 (Wed) 11:59 pm on HuskyCT (100 points)

The goal of Project 2 Part I is to help you learn how to create a NoSQL database instance using MongoDB, and interface with it using another programming language such as Python. You can read more about MongoDB at <https://hevodata.com/learn/mongodb-open-source/>.

1. Download MongoDB and install it in your local computing environment. You can download MongoDB from <https://www.mongodb.com/try/download/community>. Other options are possible.
2. Consider the example SQL database, say LIBRARY_DB, that you built in Project 1. In this Project 2 Part 1 you are building a standalone NoSQL database independent of LIBRARY_DB. This NoSQL database is called LIBRARY_NoSQL and it will include JSON files and image files for books which are not easy to store in MySQL. Note the example JSON file we discussed in class given below.

```
1  {
2    _id: "5cf0029caff5056591b0ce7d",
3    firstname: 'Jane',
4    lastname: 'Wu',
5    address: {
6      street: '1 Circle Rd',
7      city: 'Los Angeles',
8      state: 'CA',
9      zip: '90404'
10   }
11 }
```



You are given a set of JSON files for the corresponding books in HuskyCT /Project2/Data.

Image files are also provided in HuskyCT/Project2/Data, but not all books have an associated cover image.

* Note that each book has one and only one description doc in JSON and zero or one image (.png, or jpeg).

3. You now create a LIBRARY_NoSQL instance (possible using steps given below).

(a) Connect to MongoDB either from the shell, GUI, or a programming language of your choice (i.e., Python).

You may use MongoDB Compass to explore and visualize your database, but you **must** use a programming language of your choice, and potentially an external library, such as Python and PyMongo, in order to query the database. Doing so will be required for doing Project 2 Part 2 anyway.

(b) Open and read the image files as binary and JSON files as binary or text and insert them into collection as the following format:

```
{
  "name": "book_title",
  "book_id": "book id number",
  "image": BinData(image),
  "description": BinData(JSON) }
```

You should have data entered for 6 books, as shown below:

*Note there are two books titled *The Lost Tribe*. This is intentional, these are separate books, each with their own ID, image, and JSON description.

Book Title	Book ID	BinData(image)	BinData(JSON)
The Lost Tribe	B1
It	B2
Event Horizon	B3
The Age of AI	B4
A World Restored	B5
The Lost Tribe	B6

(c) Show the table view of the completed LIBRARY_NoSQL. Express the query (code) and show the outcome of running this query. (20 points). Hint: See attached Help Q&A.

(d) Show how many key value pairs are added in your LIBRARY_NoSQL. Express the query (code) and show the outcome of running this query. (20 points).

4. Now you show you can query your LIBRARY_NoSQL. (60 points)

(a) Show the name and document describing the book with ID = B1 – Express the query (code) and show the outcome of running this query e.g. JSON file.

(b) Show the total number of books which have an image saved. Express the query (code) and show the outcome of running this query.

(c) Show the name of the book which does not have a cover image saved. Express the query (code) and show the outcome of running this query.

(d) Print out all the book names in LIBRARY_NoSQL. Express the query (code) and show the outcome of running this query.

(e) Print out the name and ID of the books available in kindle, and the price of the kindle version for each. Express the query (code) and show the outcome of running this query.

(f) Show the ID of both books named “The Lost Tribe”. Express the query (code) and show the outcome of running this query.

Note: Show code here means the screenshot including only the MongoDB query portion not the entire code.

(Help docs available in subsequent pages)

Various Help Docs and Q&A

MongoDB Installation

Q: Which version of MongoDB Server should I be installing for this project?

A: Using the latest version of MongoDB Community Server is advised. Go to this URL: [mongodb.com/try/download/community](https://www.mongodb.com/try/download/community) and pick your operating system. Download the latest installer for your operating system and install it. The installation instructions for different operating systems will be different.

Q: How can I access MongoDB through command line?

A: Most MongoDB Community Server installations do not include mongosh, which is how you access MongoDB through command line. Find instructions for downloading and using mongosh here: <https://www.mongodb.com/docs/mongodb-shell/>

Accessing MongoDB

Q: How can I access MongoDB from Python?

A: Python: Use the PyMongo library [mongodb.com/docs/drivers/pymongo/](https://www.mongodb.com/docs/drivers/pymongo/).

Q: How can I display a MongoDB collection as a table?

A: Python: Use Pandas library <https://pandas.pydata.org/> or MongoDB Compass GUI <https://www.mongodb.com/try/download/compass>

Using MongoDB Compass GUI and PyMongo is Highly Recommended

MongoDB Compass (<https://www.mongodb.com/try/download/compass>)

You can use the Compass GUI to monitor database information and see settings on your server connection.

Example of tabular view in Compass GUI (data is from last semester's project):

The screenshot shows the MongoDB Compass interface. On the left, a sidebar lists the database 'Warehouse_DB' and its collections: 'books', 'warehouses', 'admin', 'config', and 'local'. The 'warehouses' collection is selected. The main panel displays the 'Warehouses' collection in a tabular view. The table has columns: '_id ObjectId', 'name String', 'address String', 'images Array', and 'description Binary'. There are 7 documents listed. The interface also shows a search bar, a filter button, and a query input field. The bottom right corner indicates '1 - 7 of 7' documents.

	_id ObjectId	name String	address String	images Array	description Binary
1	ObjectId('643ee2606c0e353c4...')	"Husky Vernon"	"123 Dog Lane, Storrs, CT"	[] 1 elements	BinData(0, 'JVBER10xLjchNCiW...
2	ObjectId('643ee2606c0e353c4...')	"Target LA"	"45 Palm Road, Los Angeles, ..."	[] 2 elements	BinData(0, 'JVBER10xLjQKJdP...
3	ObjectId('643ee2606c0e353c4...')	"TraderJoes Chicago"	"67 Snow Street, Chicago, IL"	[] 0 elements	BinData(0, 'JVBER10xLjMKJcT...
4	ObjectId('643ee2606c0e353c4...')	"Amazon NY"	"89 Gold Ave, New York, NY"	[] 1 elements	BinData(0, 'JVBER10xLjchNCiW...
5	ObjectId('643ee2606c0e353c4...')	"Bobs Phoenix"	"101 Clay Trail, Phoenix, AZ"	[] 3 elements	BinData(0, 'JVBER10xLjQKJdP...
6	ObjectId('643ee2606c0e353c4...')	"Costco Houston"	"203 Ranch Drive, Houston, TX"	[] 2 elements	BinData(0, 'JVBER10xLjQKJdP...
7	ObjectId('643ee2606c0e353c4...')	"Best Buy Miami"	"405 Sun Ridge, Miami, FL"	[] 4 elements	BinData(0, 'JVBER10xLjQKJdP...

PyMongo (<https://pymongo.readthedocs.io/en/stable/tutorial.html>)

PyMongo is a library used to access your MongoDB server using Python code. You can use PyMongo in tandem with MongoDB Compass GUI to manipulate, query, and view your database easily.

Example connection in PyMongo:

```
from pymongo import MongoClient
import os

client = MongoClient('mongodb://localhost:27017/')

db = client.Warehouse_DB
```

From here, you are connected to the database, and can begin inserting, removing, and querying entries in the collections within the database.

More information can be found in the tutorials linked here:

- <https://pymongo.readthedocs.io/en/stable/tutorial.html>
- mongodb.com/docs/drivers/pymongo/

Project 2 Part 1 Deliverable

Report Format: Your report must be a PDF document with your full name as the file name (e.g., JohnDoe.pdf). The top portion of your report must include “Project 2 Part I, Your full name, Date” for easy identification for grading purpose and the problem numbers/answers in sequence as illustrated below —Note: Penalty for not following this requirement.

Late submission penalty, **5% off per day**. **Submission cut-off** is 11/15/2023 (Wed) 11:59 pm.

Project 2 Part 1

Name: _____

Date: _____

3.(c)

3.(d)

4.(a)

4.(b)

4.(c)

4.(d)

4.(e)

4.(f)
