Open-Source Technology Use Report

Proof of knowing your stuff in CSE312

Guidelines

Provided below is a template you must use to write your report for each of the technologies you use in your project.

Here are some things to note when working on your report, specifically about the **General Information & Licensing** section for each technology.

- **Code Repository**: Please link the code and not the documentation. If you'd like to refer to the documentation in the **Magic** section, you're more than welcome to, but we'd like to see the code you're referring to as well.
- **License Type**: Three letter acronym is fine.
- **License Description**: No need for the entire license here, just what separates it from the rest.
- **License Restrictions**: What can you *not* do as a result of using this technology in your project? Some licenses prevent you from using the project for commercial use, for example.
- **Who worked with this?**: It's not necessary for the entire team to work with every technology used, but we'd like to know who worked with what.

Also, feel free to extend the cell of any section if you feel you need more room.

If there's anything we can clarify, please don't hesitate to reach out! You can reach us using the methods outlined on the course website or see us during our office hours.

Pymongo

General Information & Licensing

Code Repository	https://github.com/mongodb/mongo-python-driver/
License Type	Apache-2.0 License
License Description	● Commercial use ● Modification ● Distribution ● Patent use ● Private use
License Restrictions	● Trademark use ● Liability ● Warranty
Who worked with this?	Zaki, Chi Ho, Tenzin

Use as many of the sections below as needed, or create more, to explain every function, method, class, or object type you used from this library/framework.

MongoClient

Purpose

- The Client class in mongoDB.py uses the MongoClient object to connect to the database.
- It is used to create multiple collections of the database for us to store data.
- It is used in mongoDB.py:
    - In init function, we created 3 collections for users, chats, and posts.
    - In registerUser function, we inserted new usernames and rejected the same usernames.
    - In loginUser function, we checked if the username and password were the same to sign the user in or deny them.
- It is used in handlers.py:
    - We imported the functions so we could call the functions in the respective routes. This was to make the code more organized and readable.

Magic ★★。˚⋆˚☽˚｡⋆.˚★彡⋆༄

- We import pymongo so we can use a database to store data for users, chats,and posts. Once mongoDB.py is imported in handlers.py, it creates a Client class that creates a connection to a mongoClient and a mongo database with multiple collections in it. Each collection stores data for users, chats, and posts.
- Currently, we have a method registerUser and loginUser for users. RegisterUser will check if the given username in the parameter matches any username existing in the user collection. If so it will reject the registration, else it will successfully create the account, thus inserting the username password json object into the collection. LoginUser will check if the given username and password matches any pair in the user collection, if so it will successfully log the user into their account, else deny the user from access.
- In handlers.py, we imported mongoDB.py to use the functions registerUser and loginUser. We used registerUser in the register function for the purpose explained previously. We used loginUser in the login function for the purpose explained previously.

- https://github.com/mongodb/mongo-python-driver/blob/master/pymongo/mongo_client.py
    - Line 76 describes the MongoClient class that we use to create a connection to the database
- https://github.com/mongodb/mongo-python-driver/blob/master/pymongo/database.py
    - Line 42 describes the database class which represents the actual mongo database. In mongoDB.py, the Client class uses this to create self.db in line 9.
- https://github.com/mongodb/mongo-python-driver/blob/master/pymongo/collection.py
    - Line 68 describes the collection class that is inside of the database. In mongoDB.py, the Client class uses this to create multiple collections for users, chats, and posts in lines 10, 11, 12.
- https://github.com/mongodb/mongo-python-driver/blob/master/pymongo/operations.py
    - Line 23 describes the insertOne function that allows us to insert a json objective into the collection.
    - Line 55 describes the deleteOne function that allows us to delete a json objective from the collection.