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

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, posts, and online users.
    - 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. As well as if the password meets the requirements.
    - In toggleDarkmode function, we check if the darkmode is toggled for the user's setting and toggle it on or off accordingly.
    - In getDarkmodeStatus function, we check if darkmode is on or off and return that.
    - In getOnlineUsers function, we get all the users who are currently online.
    - In addOnlineUser function, we insert the current user into the online users collection.
    - In removeOnlineUser function, we delete the current user from the online users collection.
    - In usernameTaken function, we check if the username is used in an existing account.
    - In getAllUsers function, we get all the users that exist in the users collection
    - In mongoCreatePost function, we insert a newly created post into the posts collection.
    - In mongoGetPost function, we get the specified post by the post ID from the posts collection.
    - In mongoAddPostToUser function, we insert the given post to the respective user in the users collection.
    - In mongoGetUserPosts function, we get all the posts for the respective user in the users collection.
    - In mongoGetAllPosts function, we get all the posts from every existing user in the users collection.
- It is used in the 3 handlers (authHandlers.py, chatHandlers.py, postHandlers.py) in the

Handlers folder and app.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

- We import pymongo so we can use a database to store data for users, posts,and online users. Once mongoDB.py is imported in app.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, posts, and online users.
- In the 3 handlers files, we imported mongoDB.py to be able to insert, delete, and find one/all data into the 3 different collections in our mongo database. For example, we used the users collection to store all the accounts created. We used insertOne to insert an object of username to the given username and password to the given password. The password is hashed before being stored to provide security. Another example is when we used deleteOne for online users collection to delete a user from the collection when they log out. Another example is when we used findOne for users collection to check the collection to see if there exists an object (account) with a given username, so we don't have duplicate username accounts. Lastly, we used findOneandUpdate to find a specific object in the collection with the respective data and update with the given data.

- 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 12, 14, 16.
- 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 object into the collection.
  - Line 55 describes the deleteOne function that allows us to delete a json object from the collection.
- https://github.com/mongodb/mongo-python-driver/blob/master/pymongo/collection.py
  - Line 1103 describes the findOne function that allows us to find a json object from the collection if the object matches the given requirement.
  - Line 2464 describes the findOneandUpdate function that allows us to find a json object from the collection if the object matches the given requirement and updates the object key-value's value with the given data.