University of the Philippines Los Baños
CMSC 100 - Web Programming Laboratory Exercise

Exercise 06: MongoDB

Title: User Management using Express, Mongoose, Needle, and MongoDB compass.
Type: Individual Assessment
Score: (15 points)

Problem: Create a simple user management using Express, Mongoose, Needle, and MongoDB compass. **Use the terminal or command line interface to test your files.** No need for front-end scripting. Submit your code to Github Classroom and submit a screenshot of results in the Google Classroom. Compress the attachments as a ZIP file. Visit this link for the Laboratory guide of this exercise.

Tips:

1. You need to install the following NPM packages:
    - Express - to create a web server
    - Mongoose - to access MongoDB operations using JavaScript
    - Needle - test the POST endpoint.

2. You need to install MongoDB compass.
    - Link right here: https://www.mongodb.com/products/tools/compass
    - Documentation: https://www.mongodb.com/docs/

Program specifications:

1. Create a new database named: StudentDatabase
2. In your database, create a collection named studentData.
3. Base your Mongoose model on the following object format:
    a. Student
        - stdnum (String)
        - fname (String)
        - lname (String)
        - age (Number)

4. Create a server with the following routes and endpoints. You need to separate the components of your server into different files such as server, controller, and router.
    a. POST `/save-student`
        - Main feature of this endpoint is to save a new student in the database, if success then returns an object {inserted:true}, otherwise {inserted:false}
        - Assume that the user inputs the complete details of the student.
        - Create 5 different students including *Mary Jane Watson*.

    b. POST `/update`
        - Search for an existing student's *fname* (e.g. "Mary Jane")

        - Update the student’s *fname* using the **.updateOne()** method.
        - Update her last name to *Parker*
    c. POST `/remove-user`
        - Removes a specific user using **deleteOne** method.
        - Example: await Student.deleteOne({stdnum: ‘8051495845’})
    d. POST `/remove-all-user`
        - Removes all users using **deleteMany** method.
        - If successful, send an object {deleted:true}, otherwise {deleted:false}, check the return object of the model (acknowledge/deleteCount)
    e. GET `/user`
        - Searches user by the username. It should return an array of JSON objects containing the user. otherwise an empty array.
        - Example: http://localhost:3000/user?stdnum=‘8051495845’
    f. GET `/members`
        - returns an array of JSON objects containing all users, otherwise an empty array.

5. Do not include NODE MODULES:
    - Make sure to include the `package.json` file
    - On your terminal, type in nano .gitignore
    - Write:

```
node_modules/*
```

6. README File:
    - Project Title
    - Author
    - Code Description
    - How to use

Submission Guidelines: *Accept the invite link posted in the Google Classroom's exer 7 to create your own GitHub classroom repository. Push your files in your Github repository. Make sure to include the package.json file. DO NOT INCLUDE THE node_modules/ folder. Kindly put a .gitignore file so that the node_modules will not be pushed to your remote repository. Do not forget to document your code with comments.*

After you are done with your exercise, click the mark as done button in the Google Classroom Assignment.

Scores breakdown:

Criteria	Description
Use of GitHub (5 Points)	● Commit Messages - 3 points ● Documentation/ReadMe -1 point ● Use of GitHub classroom’s repository for GitHub Pages - 1 point

Use of Mongoose and MongoDB Compass (4 Points)	● Demonstrates understanding of Mongoose's role as an ODM library for MongoDB. ● Ability to install and set up Mongoose in a Node.js environment. ● Ability to install and configure MongoDB compass.
Routes and Endpoints (6 Points)	● Performed the required POST methods (4 points) ● Performed the required GET methods (2 points)