**ITE5315 Assignment 4**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*

* + ITE5315 – Assignment 4
  + I declare that this assignment is my own work in accordance with Humber Academic Policy. \* No part of this assignment has been copied manually or electronically from any other source \* (including web sites) or distributed to other students.

\*

* + Name: Kovid Behl Student ID: N01579154 Date: 2024-3-26

\*

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*/

**Description:** Using the concepts of week8,9,10 we are going to develop a new Node/Express app program extracting data from databases. We are going to design a simple app and using MongoDB Atlas based on the given instructions on Q1 and Q2.

**Question 1)** You are asked to develop a sample Node/Express app which interact with MongoDB database in Atlas. Complete the following steps and take screenshot of the outcome of each step:

**Step1:** Create new Node/Express app (in new folder named “Asn4-mongo-yourname”) as follow:

* Add the following dependencies to your project

A computer screen shot of a computer screen

Description automatically generated

* Make sure to have the following project structure:

|  |  |
| --- | --- |
| Create a “config” folder which contains  “database.js” (will be used for database connection parameters for Mongodb.)    Create “models” folder which contains “employee.js” (will be used to create employee schema and model.)    Create “app.js” in the project root.    Make sure to have proper project settings and dependencies in “package.json” |  |

**A computer screen with text on it

Description automatically generated**

**Step2:** Using the attach file (codenippet-mongo.txt), copy/paste the related code to “config/database.js”, “model/employee.js”, and “app.js”.

* *Note: Create a new MongoDB database in your local machine and update the “url” in database.js accordingly*

**A computer screen with text on it

Description automatically generated**

A computer screen with text on it

Description automatically generated

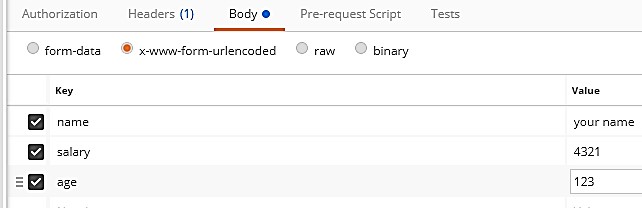
A computer screen shot of a program

Description automatically generated

**Step 3**: Run the application and test it using the following routes:

* A) Open Postman, choose POST method and use the url <http://localhost:8000/api/employees> o

o



* \ In the Body, choose “x-www-form-urlencoded” and add three pair of key values as follow o Click on send. What is the output?

A computer screen shot of a black screen

Description automatically generated

* + Check the console in VScode, what is the output?

A computer screen shot of a black screen

Description automatically generated

* + Check the Compass, is the new record added to “Employee” collection in Database?

A screenshot of a computer

Description automatically generated

* B) Open browser and enter <http://localhost:8000/api/employees>. What is the output?

A screenshot of a computer

Description automatically generated

* C) Using any \_id of any employee records in database, run the following query in the browser:
  + <http://localhost:8000/api/employees/618cf962f36b27c5379212b7>. What is the output?

A screenshot of a computer

Description automatically generated

**Step 4**: Base on your observation, answer the following questions:

1. How does the Step3:A, B,C work? Explain the work flow, route, and the way the query executed.

* A POST request to /api/employees adds a new employee to the database based on the data sent in the request body.
* A GET request to /api/employees retrieves all employees from the database.
* A GET request to /api/employees/:id retrieves a specific employee based on the provided ID.

1. What is the role of:
   1. module.exports = mongoose.model('Employee', EmpSchema);

This line exports the Employee model, which allows us to interact with the "Employee" collection in the MongoDB database using Mongoose methods.

* 1. Employee.findByIdAndUpdate

This Mongoose method is used to find an employee by ID and update their details.

1. Using the idea of Step3:C, try to update one of the record in the employee table .Find related route ☺ in the code and explain how it works.

A screenshot of a computer

Description automatically generated

* Find the related route in the code (e.g., /api/employees/:id) that handles updating an employee record.
* This route uses Employee.findByIdAndUpdate to find the employee by ID and update their details based on the data sent in the request body.

1. Using the idea of Step3:C, try to delete one of the record in the employee table. Find related route ☺ in the code and explain how it works.

A screenshot of a computer

Description automatically generated

* Find the related route in the code (e.g., /api/employees/:id) that handles deleting an employee record.
* This route uses Employee.findByIdAndDelete or Employee.remove to find and delete the employee with the specified ID

*Note:*

* 1. *It is important to explain how this app works in your video demonstration*

**Question 2)** You are asked to redesign Question 1 by using the given dataset of Assignment2.

* Step 1: Create a new MongoDB database in Atlas based on the given dataset of Assignment2.
* Step 2: Redesign the route/code in Question1 and set it up to work with product-data instead of employee data.
  + - You may need to change the “model” and routes.
    - Your app should have the following features and Demonstrate how app works using Postman/ThunderClient. (similar to Q1):
      * Show all product-info

A screenshot of a computer

Description automatically generated

* + - * Show a specific product (based on the \_id or asin)

A screenshot of a computer

Description automatically generated

* + - * Insert a new product

A screenshot of a computer

Description automatically generated

* + - * Delete an existing product (based on the \_id or asin)

A screenshot of a computer

Description automatically generated

* + - * Update “title” & “price” of an existing product (based on the \_id or asin) o Using Handlebar and Form complete the followings (*hint:use ideas from Assingment2*):

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Step 3:Using your creativity, Add a new functionality to this app.
* Step 4: Deploy the Question2 app (Cyclic)

A screenshot of a computer

Description automatically generated

* *Note:* o *It is important to explain how this app works in your video demonstration.*

**Question 3)** Open the attached sample JS file. This program use setTimeout() to simulate a running two tasks in asynchronous way. Complete the following steps

* Step 1: and run it using nodemon. Look at the output of the program.

A computer screen shot of a program

Description automatically generated

* Step 2: What if you remove wait from Task1, any error? Explain what have you learned.

If you remove the await keyword from Task 1, where an asynchronous operation is performed it will no longer wait for the operation to complete before moving to the next line of code. Instead, it will continue executing the subsequent lines of code immediately.

* Step 3: What if you remove all await/async from the task1 and 2. How do you explain changes in the output compare to Step 1?

If you remove all await and async keywords from Task 1 and Task 2, it means that the functions are no longer asynchronous and will execute synchronously, blocking the event loop until the operations are completed.

* Step 4: Bonus: Can you design the given functionality/program using Promise?

Assignment Submission:

* Add the following declaration at the top of .js files

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*

* + ITE5315 – Assignment 4
  + I declare that this assignment is my own work in accordance with Humber Academic Policy. \* No part of this assignment has been copied manually or electronically from any other source \* (including web sites) or distributed to other students.

\*

* + Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Student ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*/

* Compress (.zip) the files in your Visual Studio working directory (this is the folder that you opened in Visual Studio to create your client side code).

Important Note:

* Submitted assignments must run locally, ie: start up errors causing the assignment/app to fail on startup will result in a **grade of zero (0)** for the assignment.
* **LATE SUBMISSIONS for assignments**. There is a deduction of 10% for Late assignment submissions, and after three days it will grade of zero (0).
* Assignments should be submitted along with a video-recording which contains a detailed walkthrough of solution. Without recording, the assignment can get the maximum of 1/3 of the total.