ITC5315: Project Status Report WK 1

Created by: Malika Gupta

Student ID: N01581424

Date: April 7th 2024

# Table of contents

[Table of contents 1](#_Toc736272803)

[Step 0 2](#_Toc2003563819)

[Step 1 3](#_Toc920770620)

[Step 2 5](#_Toc411941608)

[Summary 6](#_Toc433382319)

## Step 0

Describe why you selected to use a git repository or not.

Used git repository – Downloaded and used GitHub desktop as I still need to understand the CLI more, I used the GUI. Created a main branch and week1 branch. Pushed changed from Week1 to the main branch as I progressed. It was easier to use git compared to creating backups on a hard drive and the fact that it was stored on git and not locally made it safer as my dog couldn’t have eaten it.

## Step 1

Show a screenshot Atlas/Compass once the data has been loaded. Use the Mongosh terminal to run the command: db.<collection>.countDocuments() on the “restaurants“ collection in the “5315-project” database. Show a screenshot of the result. (see <https://www.mongodb.com/docs/manual/reference/method/db.collection.countDocuments/>)

( I have entered and deleted a few documents in the restaurants collection for testing )

A screenshot of a computer

Description automatically generated

## Step 2

Use “npm run start” to start your server. Show a screenshot of it being started using this command.

1. npm run start

A screenshot of a computer

Description automatically generated

Explain any challenges or new learnings that you encountered when implementing the routes.

Challenges faced :

Kept encountering an error : “Handlebars access has been denied because ‘name’ or ‘address’ etc is not a property of its own …” when I was trying the display the restaurant information received from the database call:

(getAllRestaurants function in database.js and restaurantDisplay.hbs for the route app.post('/api/restoform',..)

Used this internet resource and added .lean() when reading the information received from Mongoose so that the json object is readable by the handlebars.  
( <https://stackoverflow.com/questions/59690923/handlebars-access-has-been-denied-to-resolve-the-property-from-because-it-is> )

Use Postman to test all the routes. Make sure to include tests for errors / invalid data.

1. **app.get('/api/restaurants/:id',..)**

Testing successful route to get all restaurant data from db based restaurant\_id:

A screenshot of a computer

Description automatically generated

Testing invalid data route to get all restaurant data from db based restaurant\_id :

When the id does not exist:

A screenshot of a computer

Description automatically generated

When the id is not a valid ObjectId:

A screenshot of a computer

Description automatically generated

1. app.put('/api/restaurants/:id',…)

Route to update an existing restaurant info based on the id

Invalid Id test – the id passed in url is not a valid ObjectId:

A screenshot of a computer

Description automatically generated

Invalid Id test – the id passed in url does not exist in database:

A screenshot of a computer

Description automatically generated

Valid test: changing the name and other restaurant details for the restaurant with id 5eb3d668b31de5d588f429f8 by passing the changes in the Body as url form encoded parameters

A screenshot of a computer

Description automatically generated

1. app.delete("/api/restaurants/:id",..) : Route for deletion of an existing restaurant based on \_id as route parameter

Valid test for deleting with an existing and valid ID

A screenshot of a computer

Description automatically generated

Invalid Id test – Id does not exist:

A screenshot of a computer

Description automatically generated

Invalid Id test – Id is not a valid ObjectId:

A screenshot of a computer

Description automatically generated

1. app.get('/api/restaurants',..) Route to get all restaurant data from db based on page, perPage and borough(optional) – Without UI – using Postman

Valid test with valid params used:

A screenshot of a computer

Description automatically generated

Invalid test - invalid param used for :

* page

A screenshot of a computer

Description automatically generated

* perPage

A screenshot of a computer

Description automatically generated

* borough:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. app.post('/api/restaurants',…) Route to add a new restaurant

Valid Test:

A screenshot of a computer

Description automatically generated

Invalid Test - Name/Cuisine/Restaurant\_id or Borough is not provided:

A screenshot of a computer

Description automatically generated

List the response codes that you used and why you used them. Show screenshots of all your tests.  
Codes used:

* 404 : Not found – when the parameters entered are valid but they are not found in the database.
* 400 : Validation errors – Parameters entered are not of the expected data type.
* 200/201: Successful execution/data retrieval
* 500 : Internal server error / Database error

Step 3

Describe the steps you used to create the UI. Describe any challenges or new learnings that you encountered while developing the UI. Include screenshots of the UI and screenshots of Postman being used to test the new route.

Used the same template as assignment 4 for the UI.

Used express handlebars and created a main.hbs in views/layouts for the main layout of all html pages.

Used a style.css sheet inside public/stylesheets for the styling of all html pages.

The same challenge encountered - .lean() to be used when retrieving data from mongoose in JSON format.

UI Routes

app.get('/api/restoform',..) Route to render the form for user to enter page, perPage and borough

app.get('/api/restoform,..) Route to get all restaurant data from db based on page, perPage and borough(optional) – **Using UI**

Valid Test:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Valid Test: Without borough

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Invalid test: Empty Page or perPage

A screenshot of a computer

Description automatically generated

Invalid Test: Invalid Borough – does not exist

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Using Postman to test UI routes:

A screenshot of a computer

Description automatically generated

## Summary

Summarize your experience while working on this project for the first week. Do you have questions about the project for the next class? Questions you want to ask your peers or professor? Would you have done anything differently? Did anything take shorter/longer than you expected?

It was time consuming to debug the errors as I encountered small errors, and it took me a bit to figure them out. Also had to figure out how to use the proper response status and error handling and make it uniform across all routes.

Have question to ask the professor why the Postman output for UI post route test is not working as expected but works perfectly in browser.