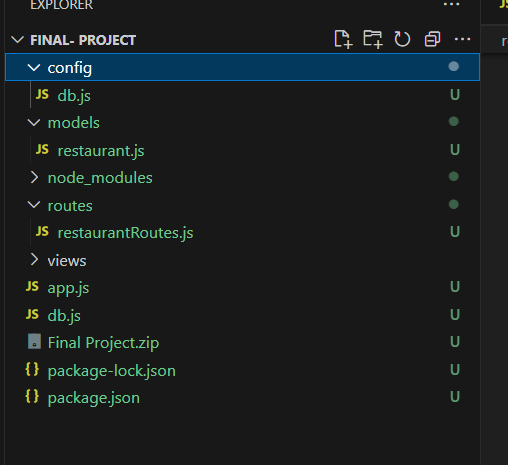
**Project-Phase-2**

**https://github.com/Punisher22/WebProgramming-Project**

**Implementation of Step2:**

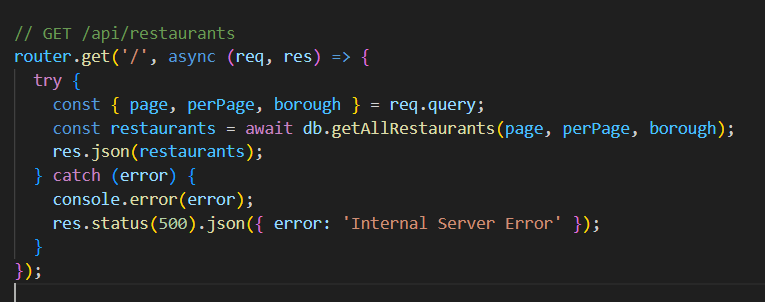
**Modified Project Structure for the phase-1**

****

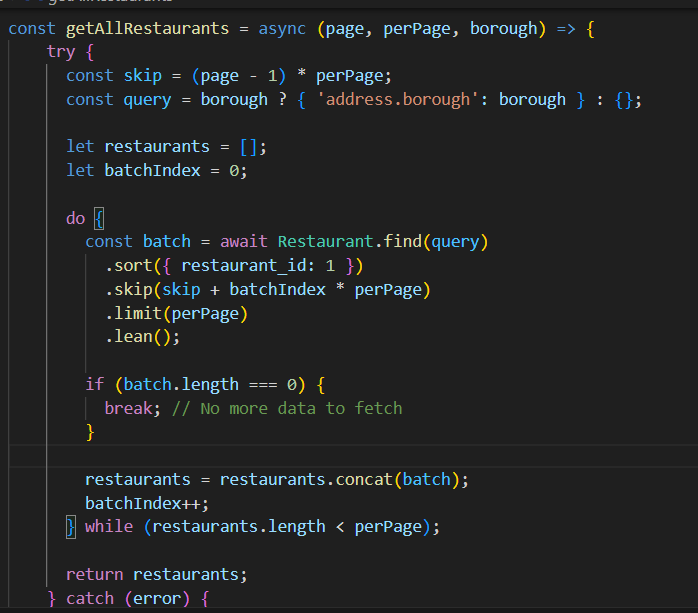
1. Db.js - has all the defined modules
2. Restaurantroutes.js - has all the defined routes
3. App.js - has connection string, db initialization
4. Models- has defined schema named as restaurant.js

**Exploring all the routes:**

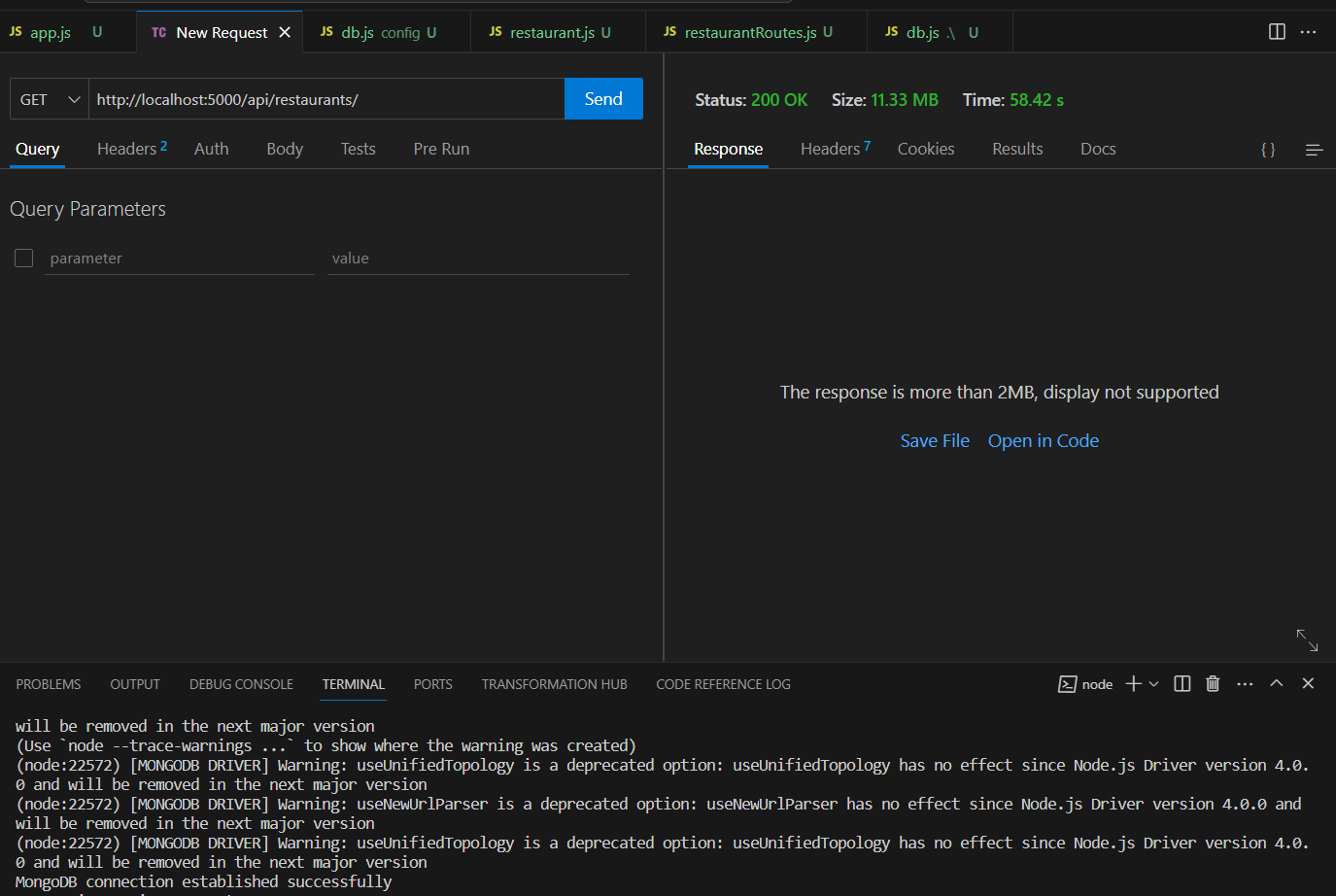
1. GET all the restaurants data by using http://localhost:5000/api/restaurants/



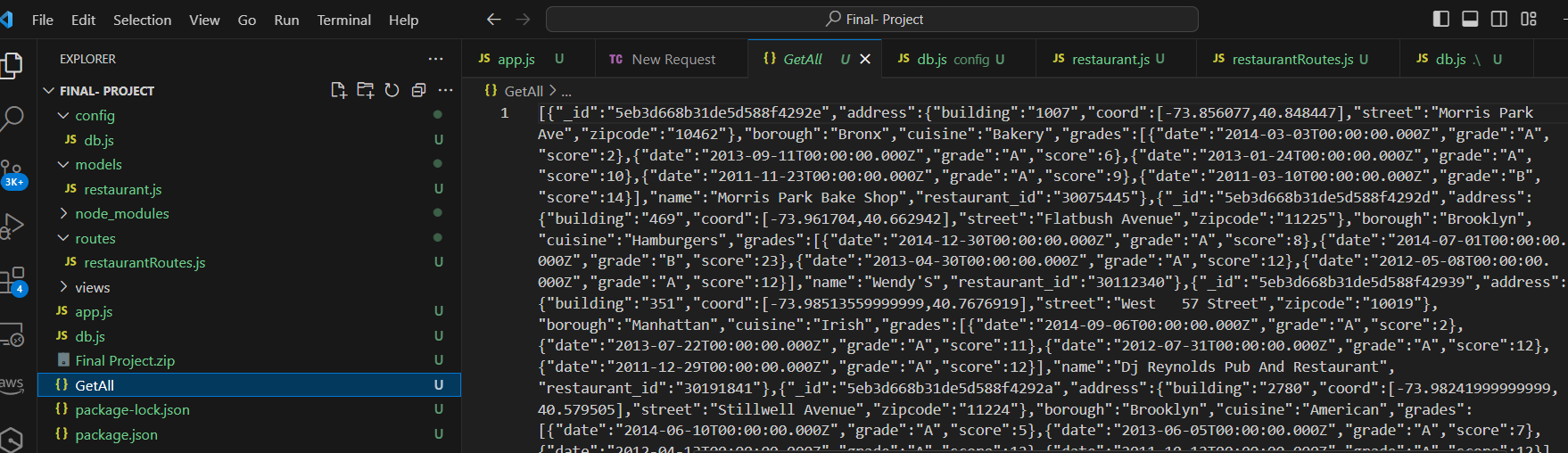
Below is the logic behind the getAllRestaurants module as we have used indexing to parse the whole data because it has a lot of restaurant data. As we run the query we have used the batch to sort the output according to id, page and per page.



As you can see below, the call is successful, but the response is large so the display is not supported so getting the data in a file.

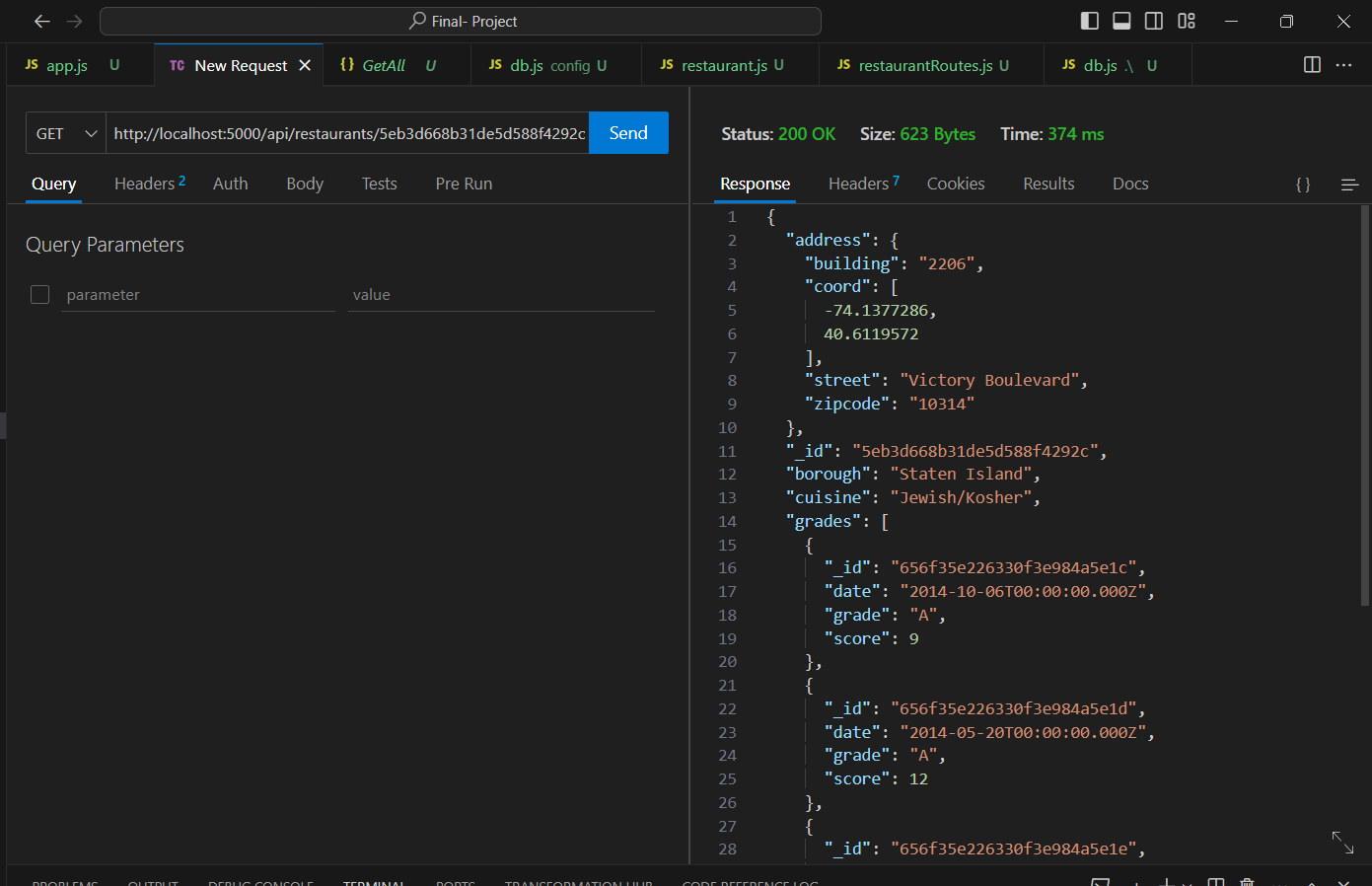


As you can see below I have saved all the restaurants in one file GetAll.txt

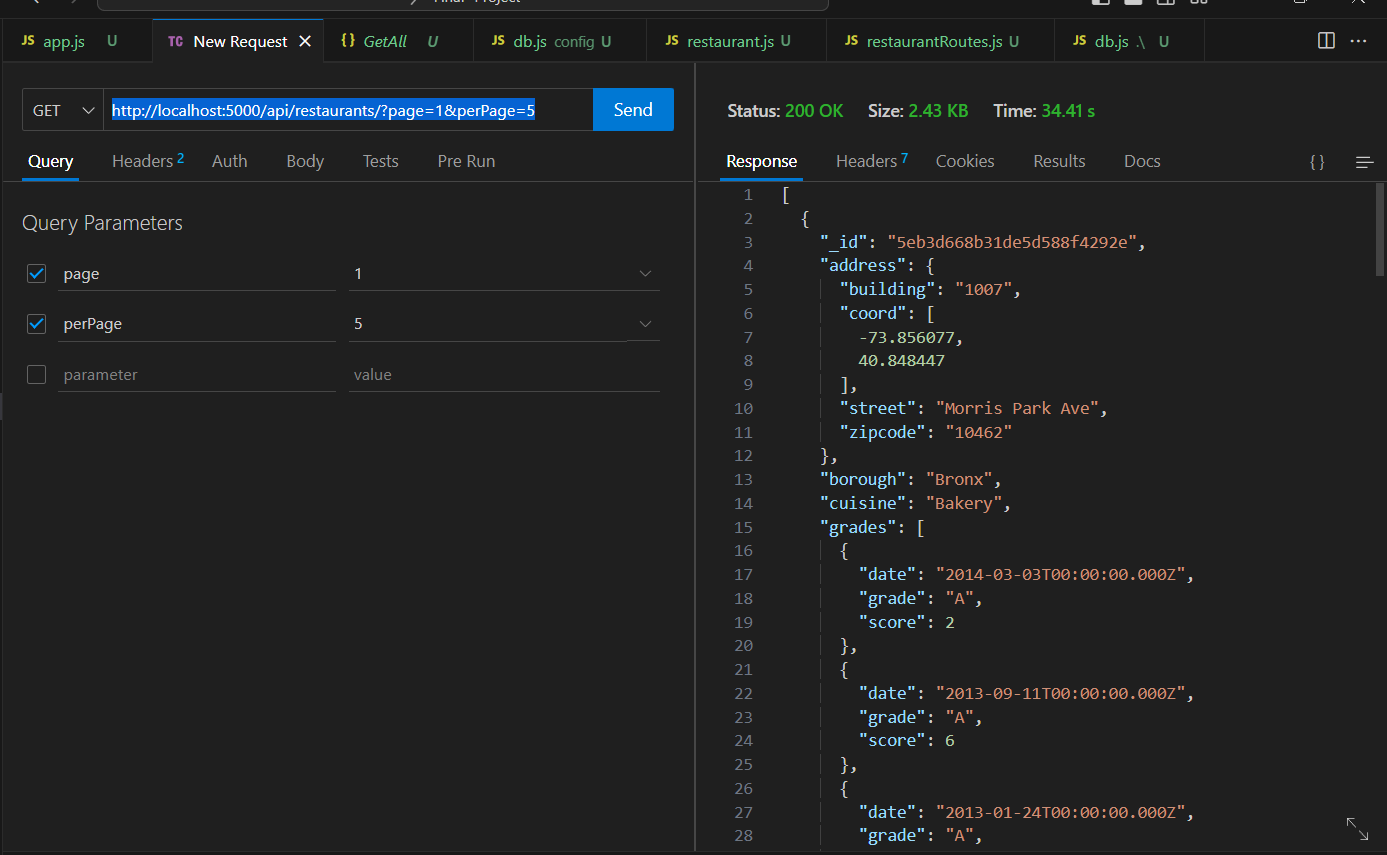


1. GET all the restaurants data by \_id using http://localhost:5000/api/restaurants/5eb3d668b31de5d588f4292c

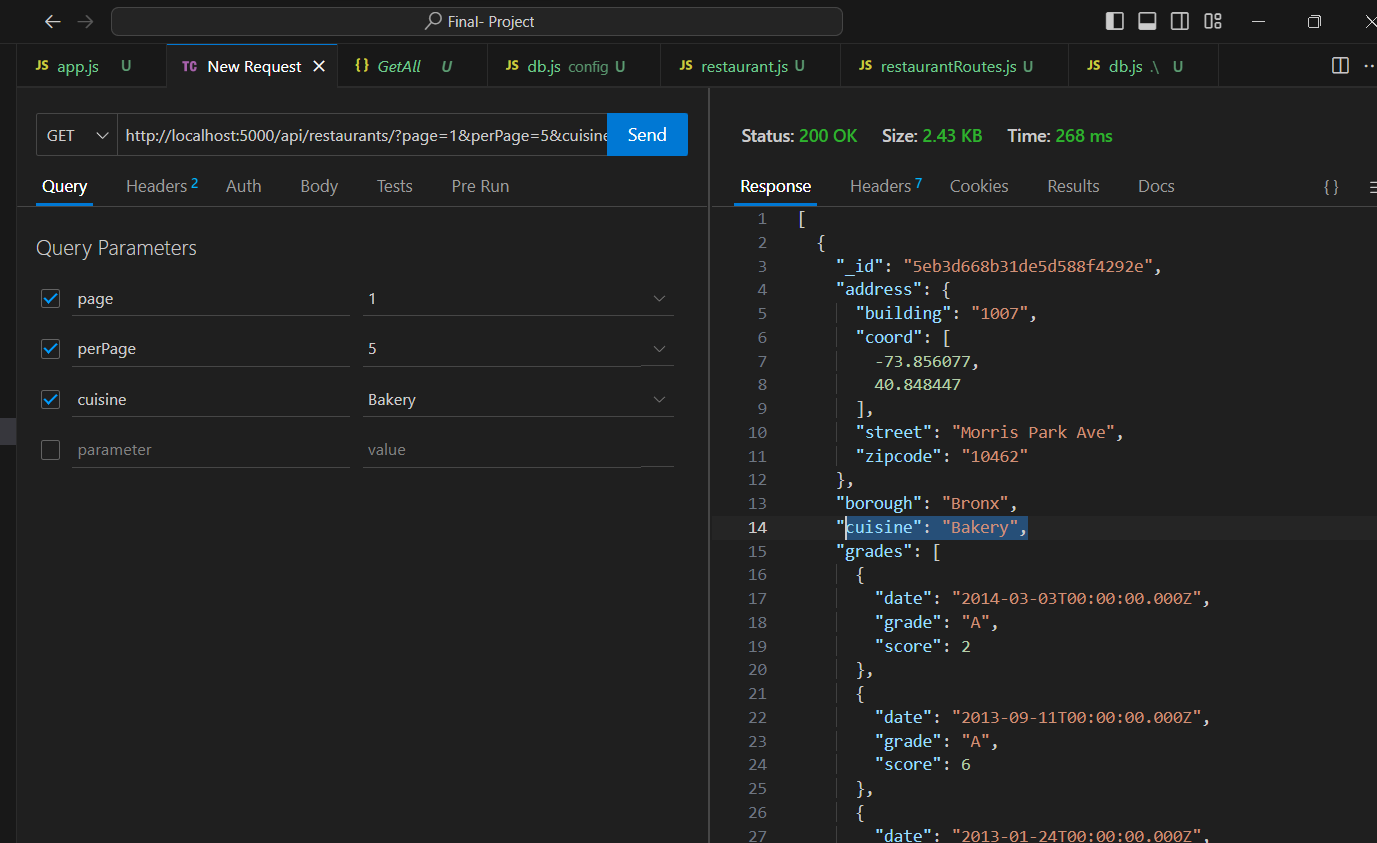
Here I’m using any random \_id and below is the response for the particular id



1. GET all the restaurants data by ?page=&perPage= using http://localhost:5000/api/restaurants/?page=1&perPage=5



Same way for optional cuisine I have passed bakery for the value and i get the data for that value as well.



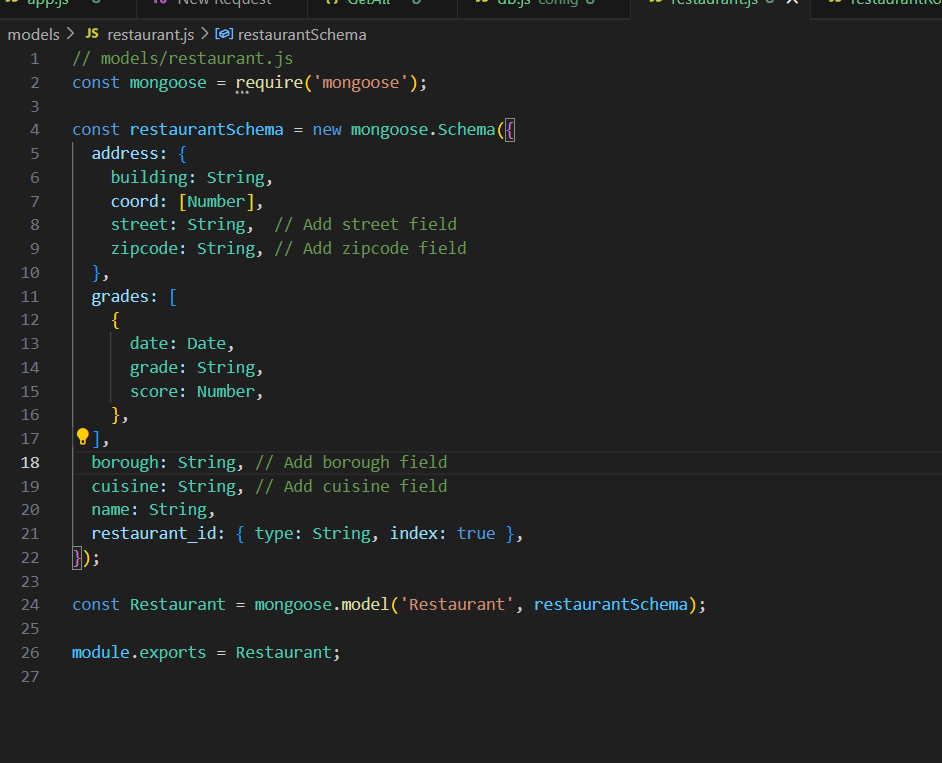
**Now POST Route:**

1. POST

We have used the addNewRestaurant module for this POST route to add the new data.



It will use below models/restuarants.js to add the data , we have defined schema here

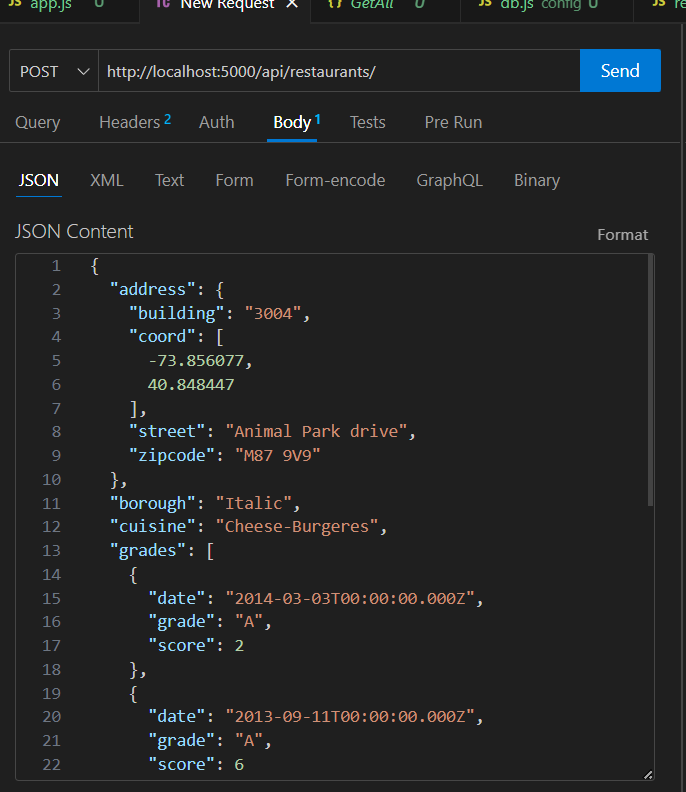


Post Route in the Restaurantsroute.js file

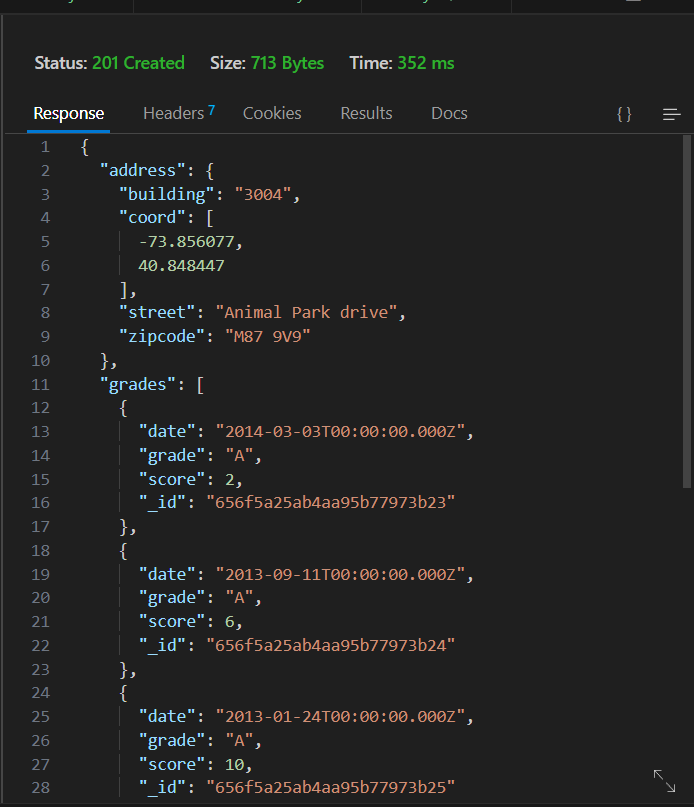


POST

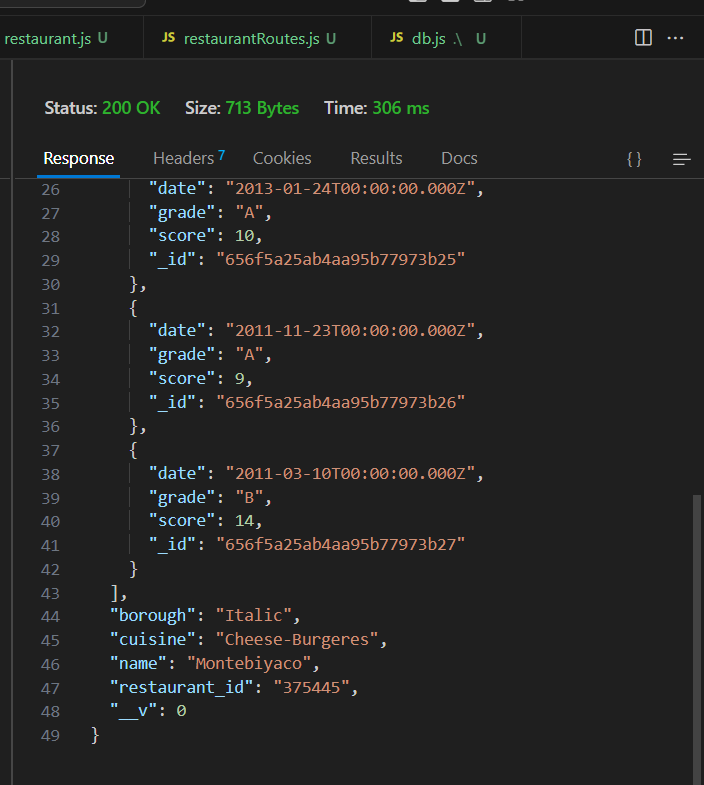
By using POST We’re adding Indian restaurant data into the json format.



I have italian restaurant data



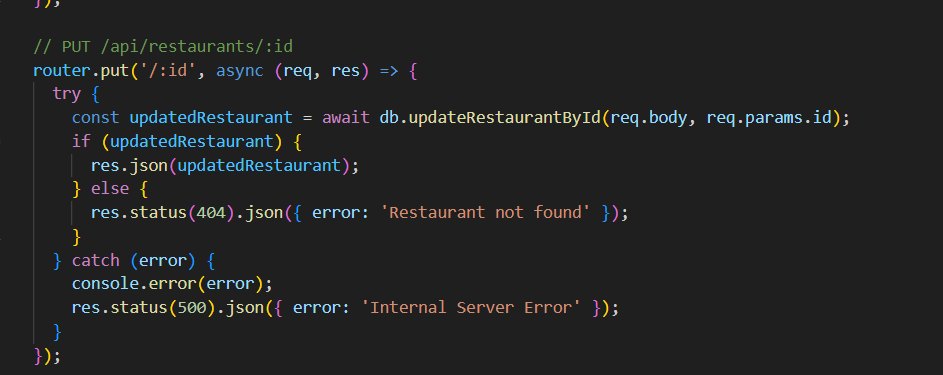
Data has been added successfully



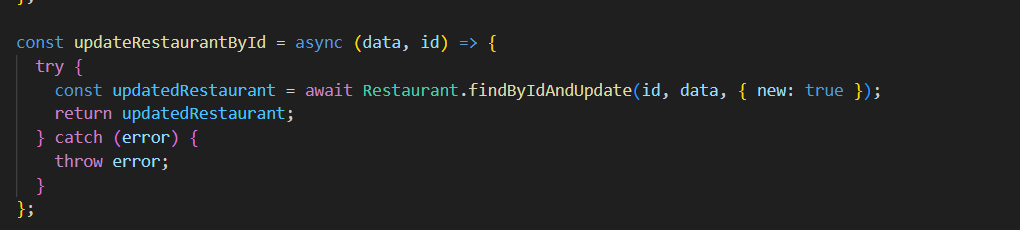
1. PUT:

<http://localhost:5000/api/restaurants/5eb3d668b31de5d588f4292a>

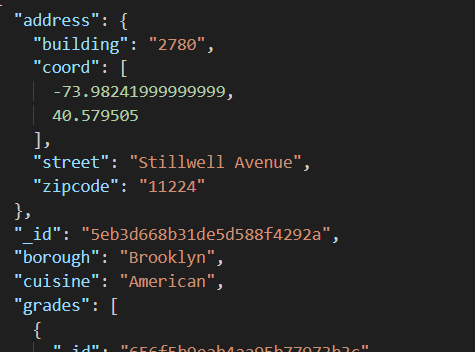
Here is the route function for the updating data that will call db.updateRestaurantbyId



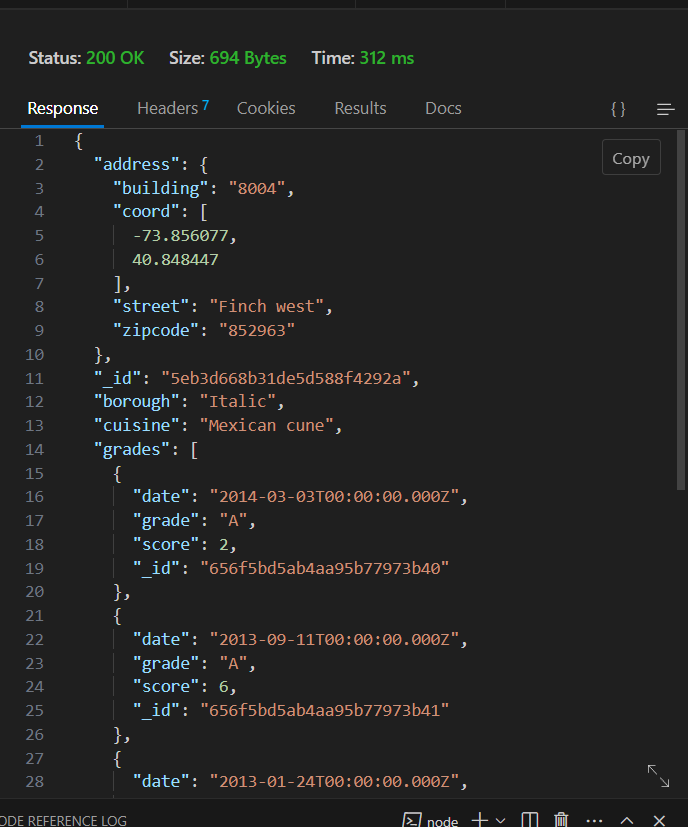
Here it will take body and update by id



For the below data we’re updating Building number, street, zip code , and cuisine

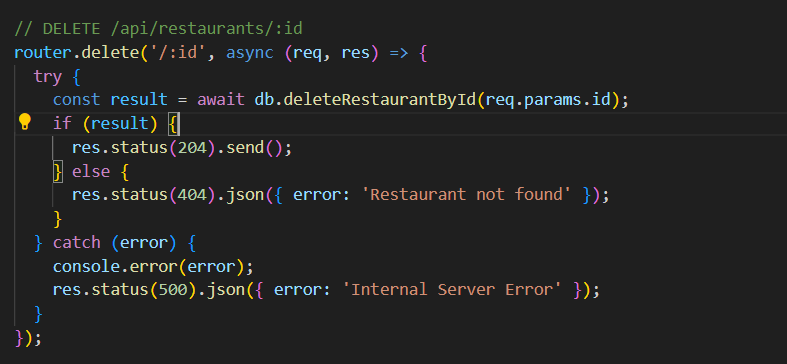


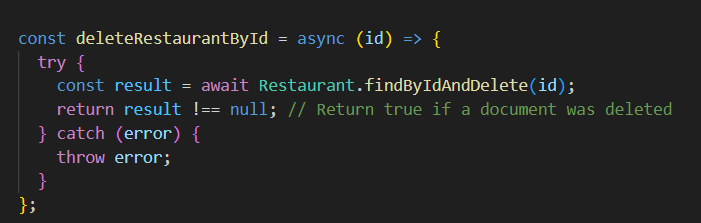
Data has been updated successfully.

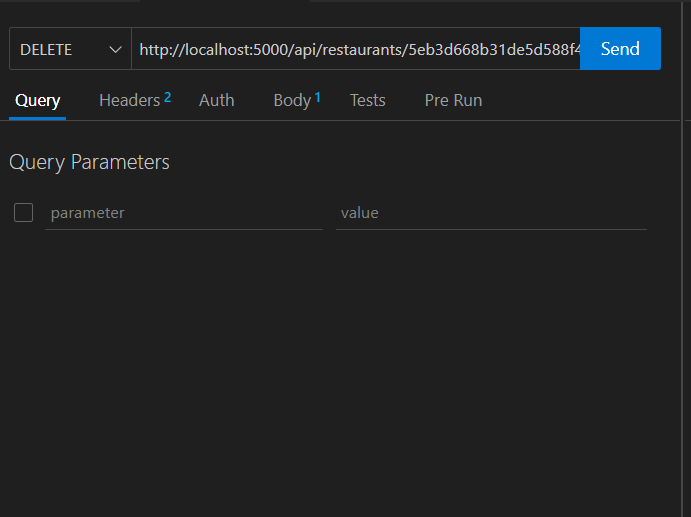


1. DELTE: <http://localhost:5000/api/restaurants/5eb3d668b31de5d588f4292a>  
   Using DELETE by \_id deleting the above updated record.

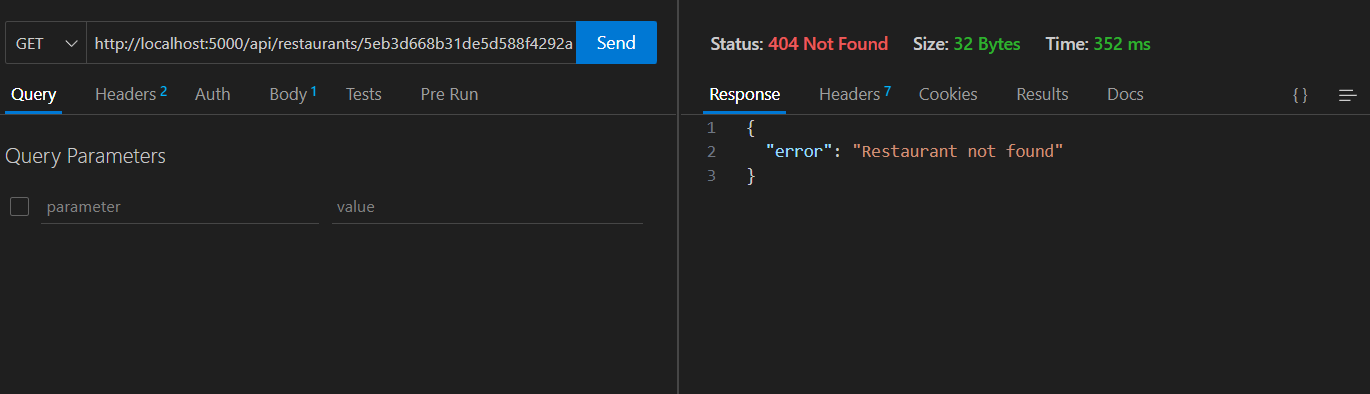
This is the delete route that will call db.deleterestaurantbyid



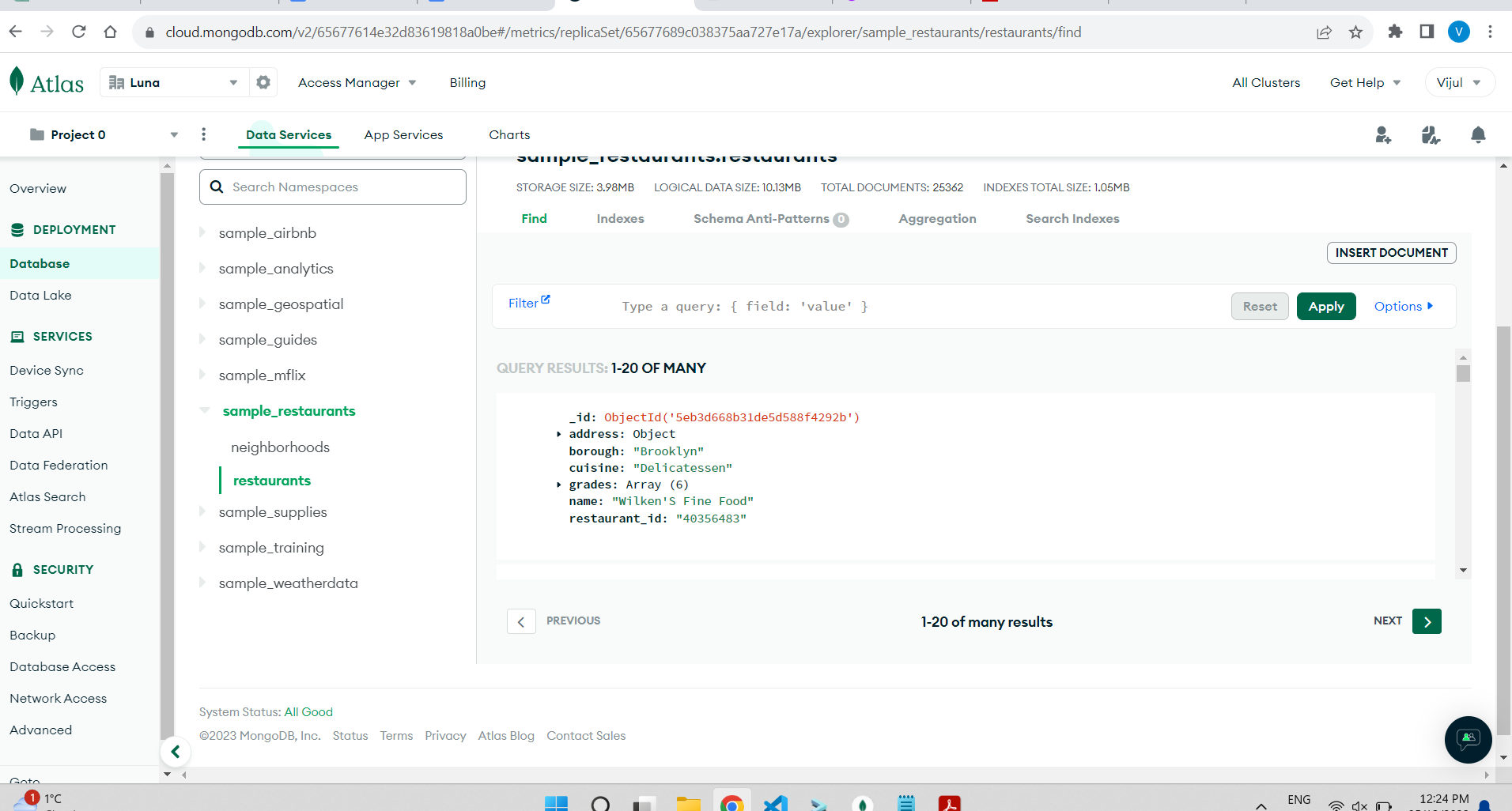




Record deleted successfully not found for the mentioned id

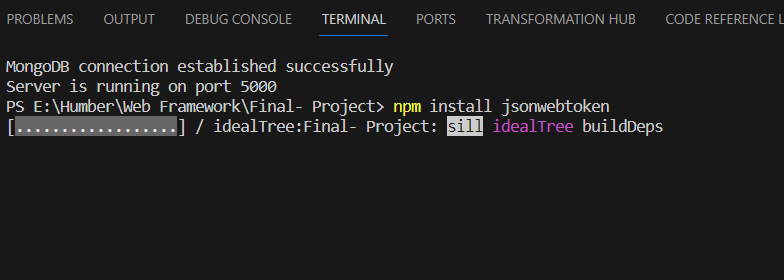


Checked Database record deleted successfully.

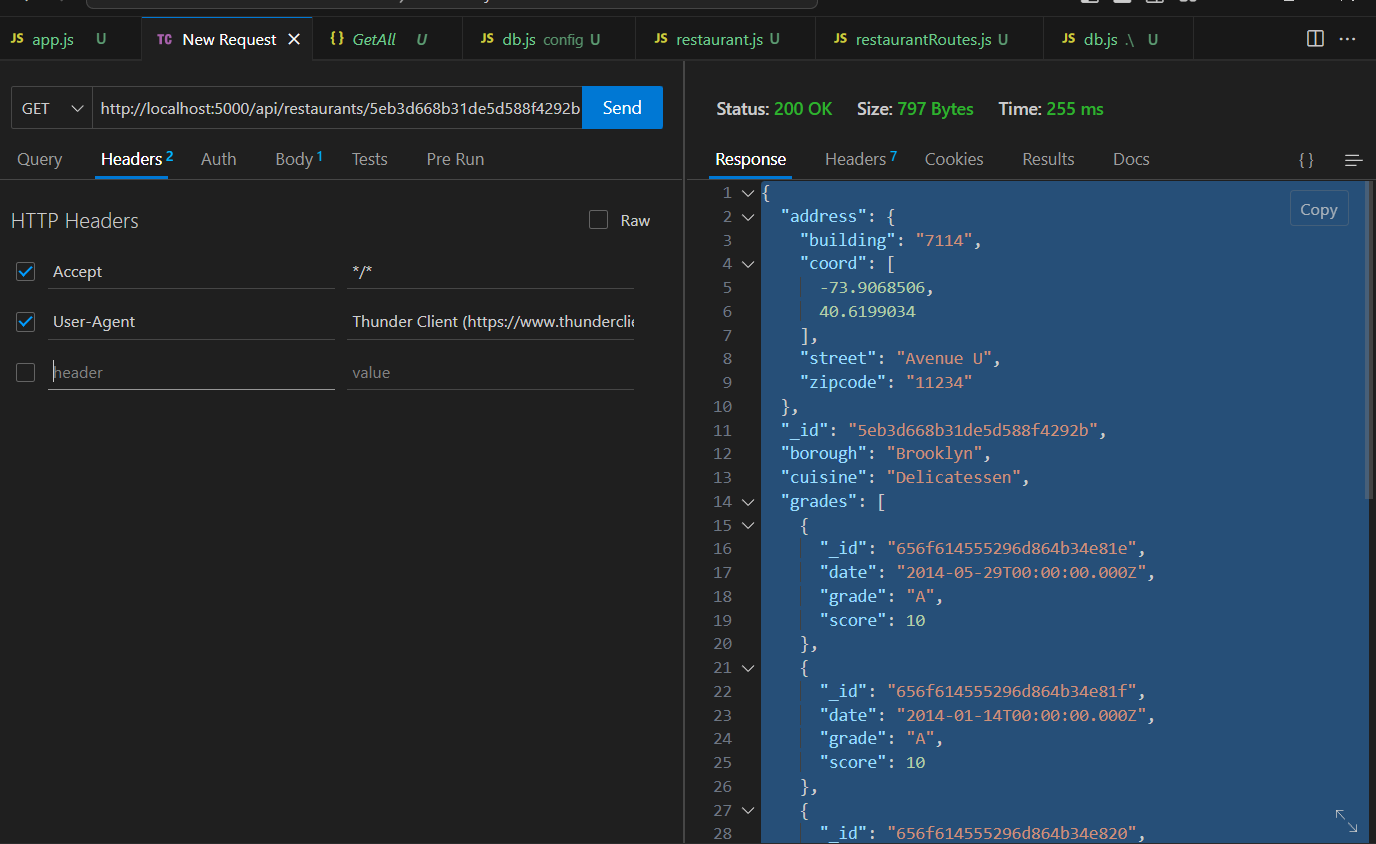


**Security Part:**

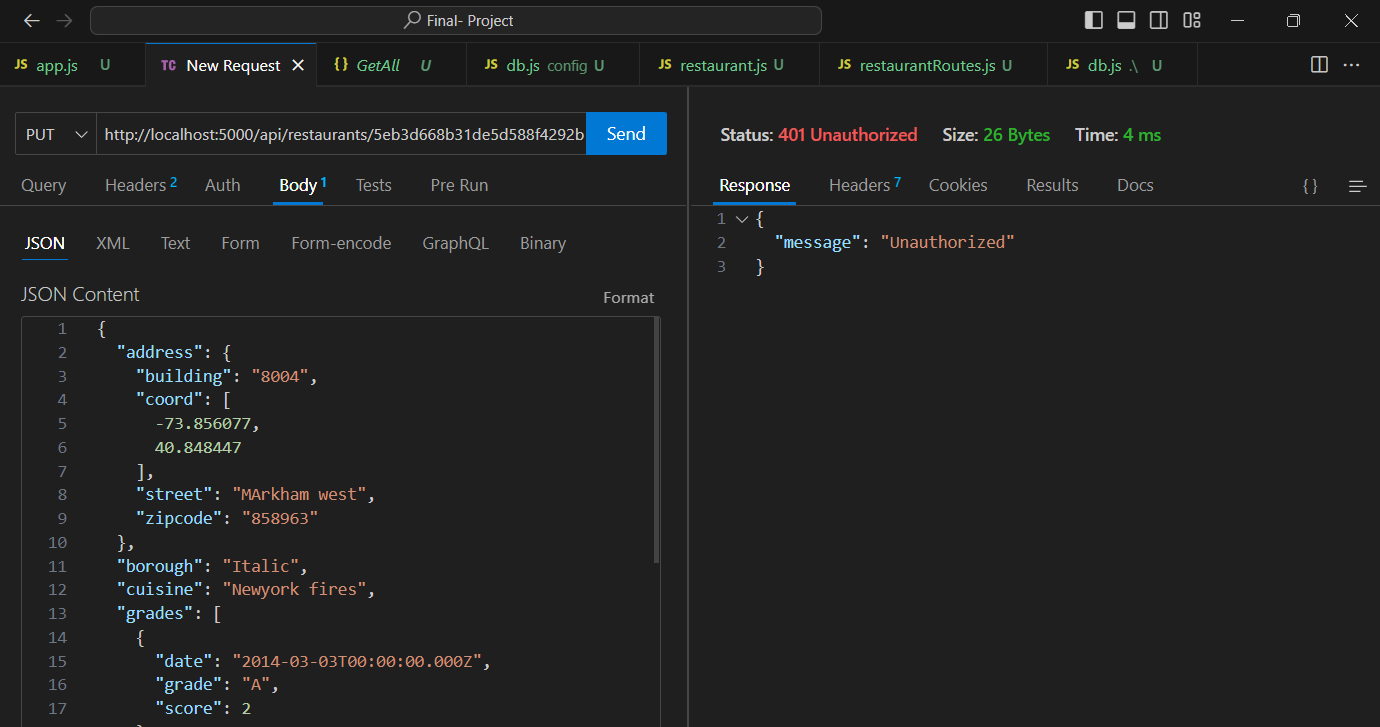
**We have added JWT Token authentication while making POST and PUT request.**



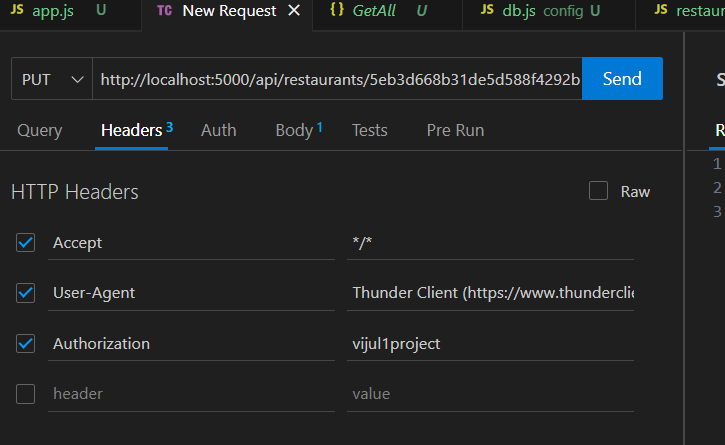
For this ID we’re making PUT request First without token



See it clear says unauthorized ,



Now making call by passing the Token and we get the updated values.



**Phase2 Implementation:**

**Created views:**

A screenshot of a computer

Description automatically generated

Display.hbs

A screenshot of a computer program

Description automatically generated

Index.hbs

A screenshot of a computer program

Description automatically generated

Login.hbs

A screen shot of a computer

Description automatically generated

Update.hbs

A screen shot of a computer program

Description automatically generated

We have implemented A UI

A screenshot of a computer

Description automatically generated

Here by using keywords U can search a restaurant

A screenshot of a menu

Description automatically generated

If user want to check all restaurants then by clicking on see restaurants they can access it.  
A screenshot of a computer

Description automatically generated

Now Searching by Name:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Contact us page to contact:

This will lead to WhatsApp account to contact the team.

A screenshot of a contact us

Description automatically generated

Login to Update features:

So, for this feature it’s a combination of the Special feature + Security Part

We’re storing the session in the form of cookie that is valid for 1 hour.  
  
and if by mistake user enter the wrong data it will directly show and error that username and password is not valid.

And once it click on the Update restaurants the data will be updated in the db as well.  
  
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

See below it’s shows that data updated successfully.

A screenshot of a computer

Description automatically generated

Updated in the database as well.

A screenshot of a computer

Description automatically generated

Here you can see that when user enter the false credentials it’s giving the Authentication failed error  
  
A screenshot of a computer

Description automatically generated