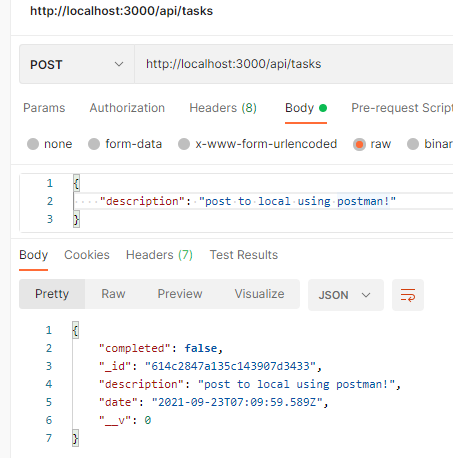
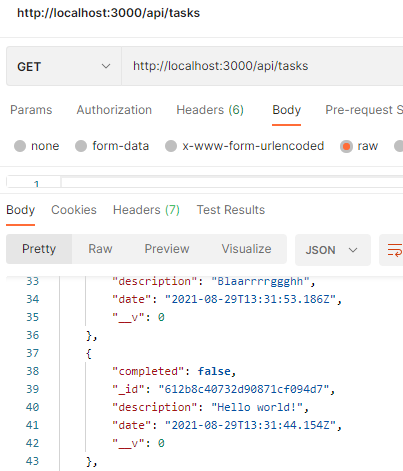
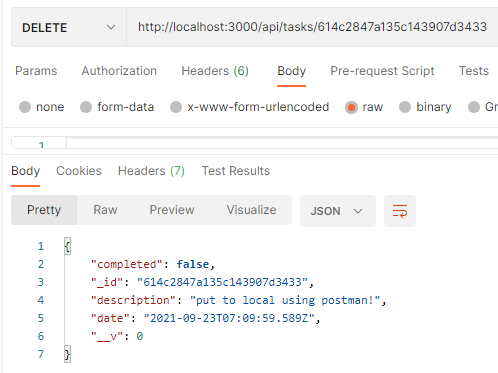
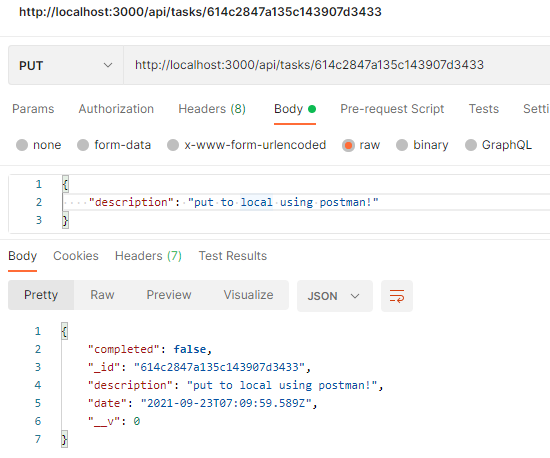
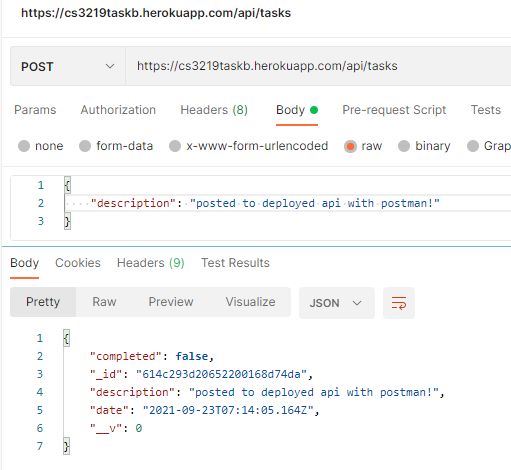
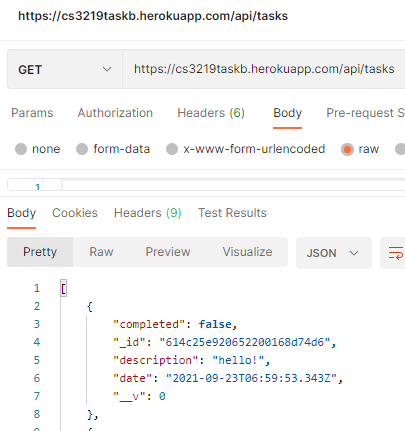
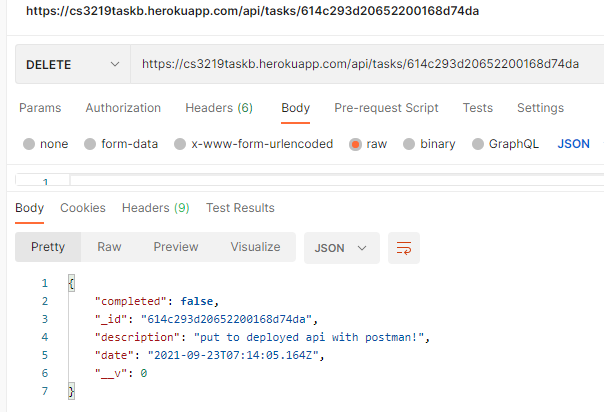
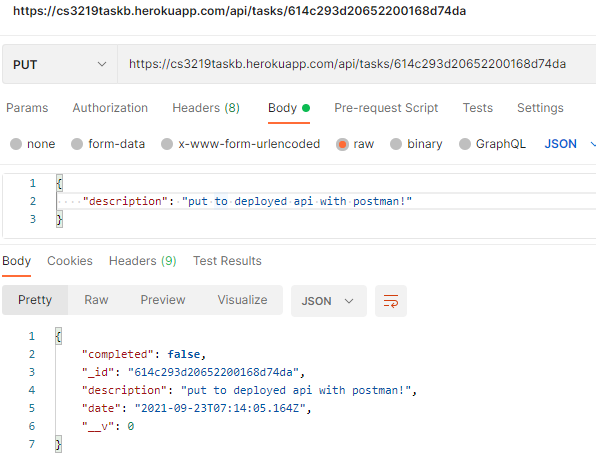
1a. Ambrose Liew Cheng Yuan, A0204750N

1b. <https://github.com/MorningLit/OTOT_Task_B_Backend>

1ci. To run the API locally, download a copy of my repository to your local machine. Next, change directory into the root folder(not inside the folder “frontend”!) with my package.json and run “npm install”, to install all the needed node module dependencies. Next, once you are done installing, run “npm run start”, this starts the API locally, or you can run “npm run server” to start the server with nodemon.

Postman calls are demonstrated with the help of the following screenshots. To localhost,  
  


**To deployed API,**  
  


**Testing Edgecases**POST but with no body.



PUT but with invalid id.  


DELETE but with invalid id.

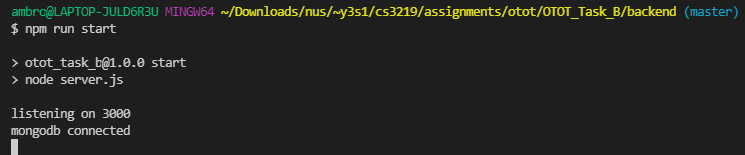
1cii. To Access the deployed API, it is found hosted on Heroku. You can perform the CRUD operations on this weblink <https://cs3219taskb.herokuapp.com/api/tasks> .  
You can perform a GET request on <https://cs3219taskb.herokuapp.com/api/tasks> .  
You can perform a POST request on <https://cs3219taskb.herokuapp.com/api/tasks> . However, you must send a json body as an object containing “description” as the key and a value for it, to perform a successful POST request to the API.  
You can perform a PUT request on   
<https://cs3219taskb.herokuapp.com/api/tasks/><<YourObjectID>> . However, you must send a json body as an object containing “description” as the key and an edited value for it and the object you are trying to edit’s ID must be found in the URL where you replace “<<YourObjectID>>” with your object’s ID, to perform a successful PUT request to the API.  
You can perform a DELETE request on <https://cs3219taskb.herokuapp.com/api/tasks/><<YourObjectID>> . However, you must have the object ID you are trying to delete in the URL, where you replace the object ID you are trying to delete with “<<YourObjectID>>”, to perform a successful DELETE request to the API.

1ciii.   
To run our tests locally, download a copy of my repository to your local machine. Next, change directory into the root folder(not inside the folder “frontend”!) with my package.json and next, run “npm install”, if you haven’t installed the required node modules. Once done, run “npm run test”, to run the tests locally. Hypothetically speaking, after a few seconds, you should be able to see the test’s output. However, you do not have my .env file which contains my MONGO\_URI environment variable as it is private and not supposed to be shared. Thus, you will see an error message.

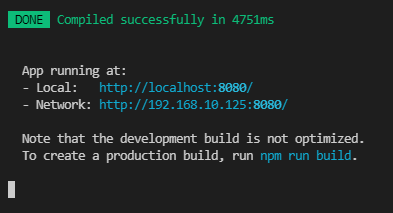
In my repository, it has the travis.yml configurations to help with continuous integrations in helping to run and check the test cases. To run our tests via travis, one way we could do is to push any changes of the repository to the GitHub repository and travis would automatically build and test for our backend application. Another way we could do is to login to <https://app.travis-ci.com/> with my credentials and force a “Restart Build” to trigger our travis to build again and run the testcases. However, again, you do not have my travis credentials and access to my GitHub account, so you will not be able to test any of my tests via travis.

1civ. To set up the frontend, change directory into the folder, “frontend”. Next, run “npm install” to install all the needed node module dependencies. Next, once you are done installing, run “npm run serve” to start up our frontend locally. You can view the frontend on <http://localhost:8080/> on your browser.

1d. Starting the local backend server’s output



Starting the local frontend server’s output



The frontend website

