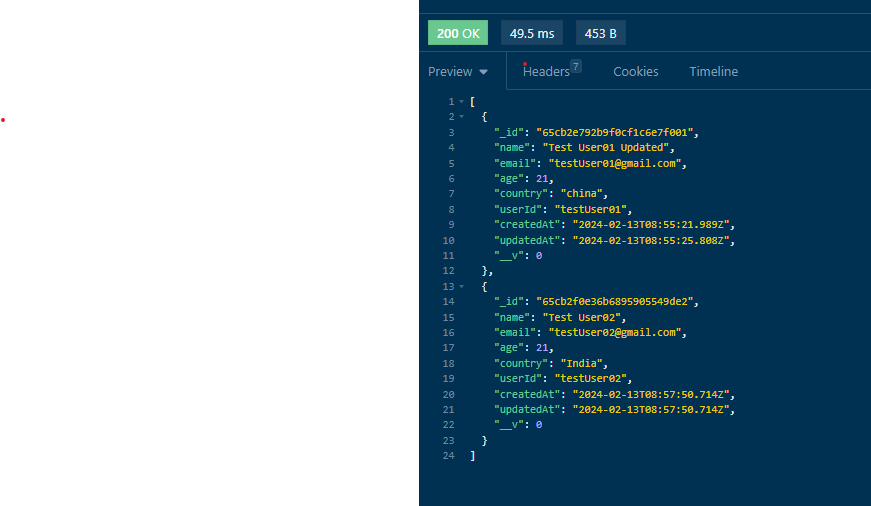
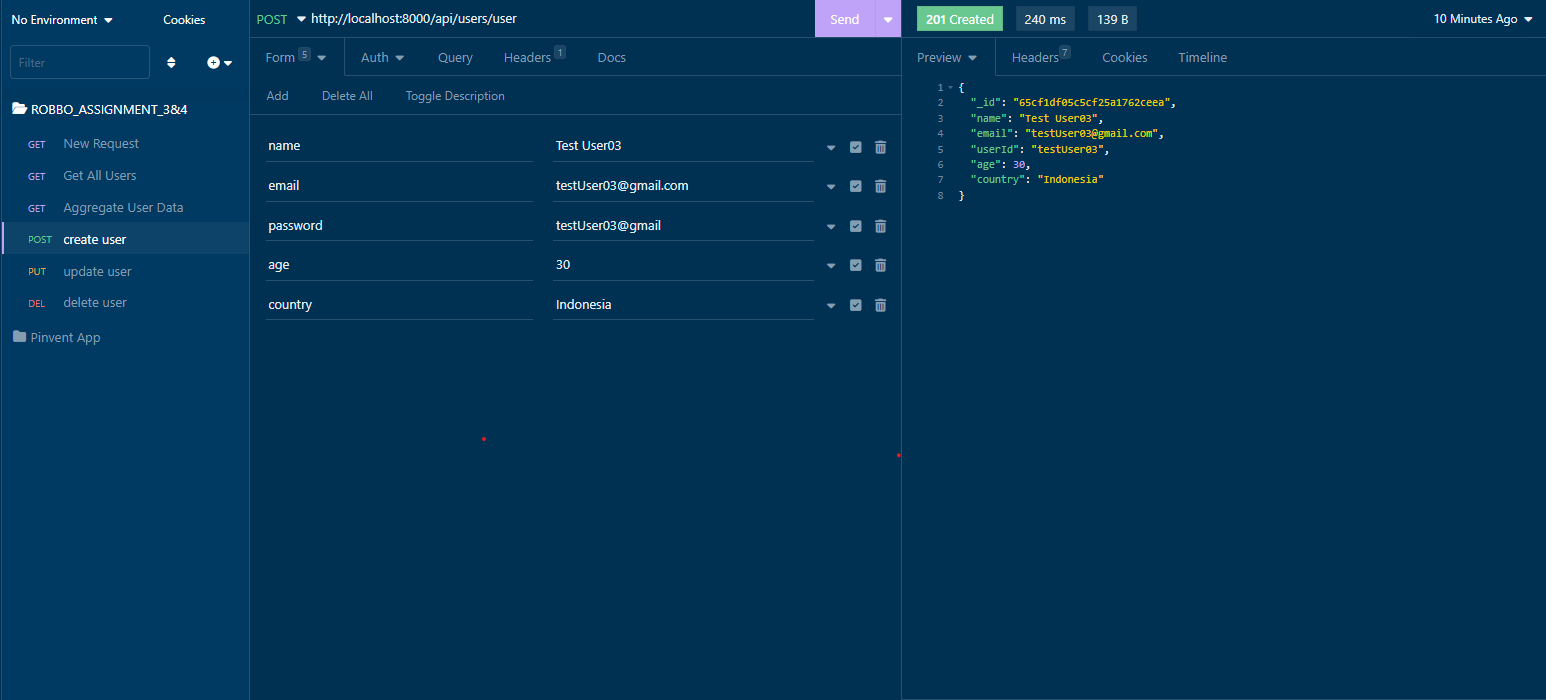
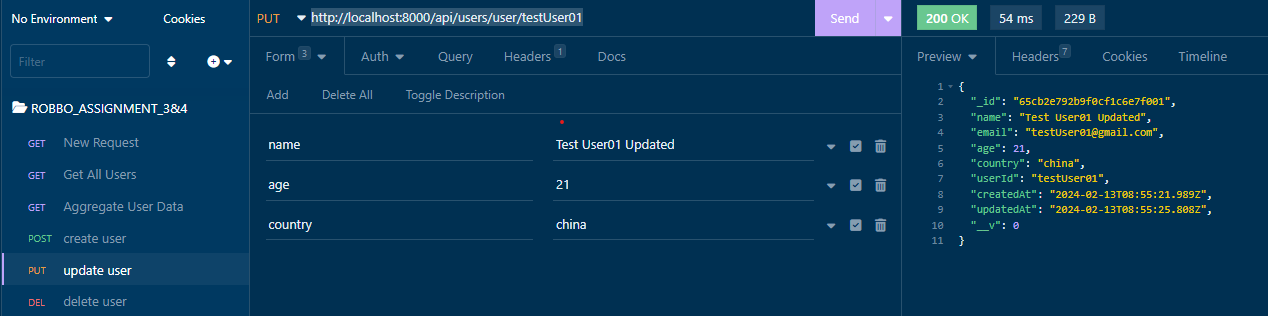
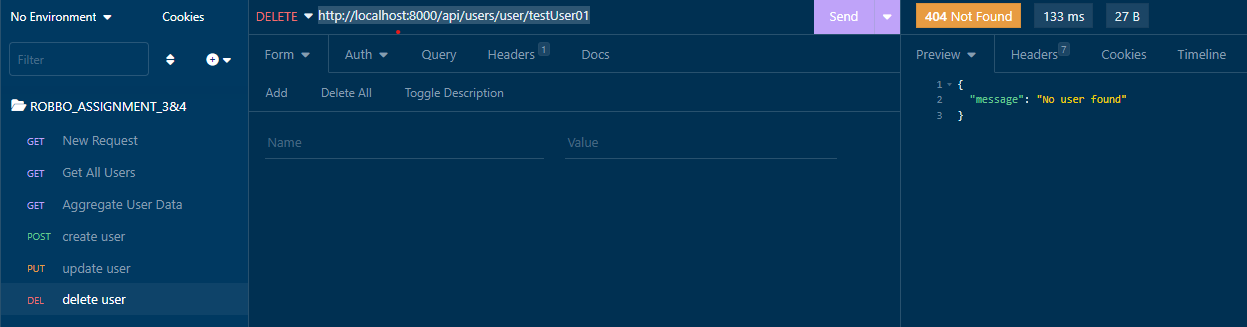
# How To Check the Assignments:

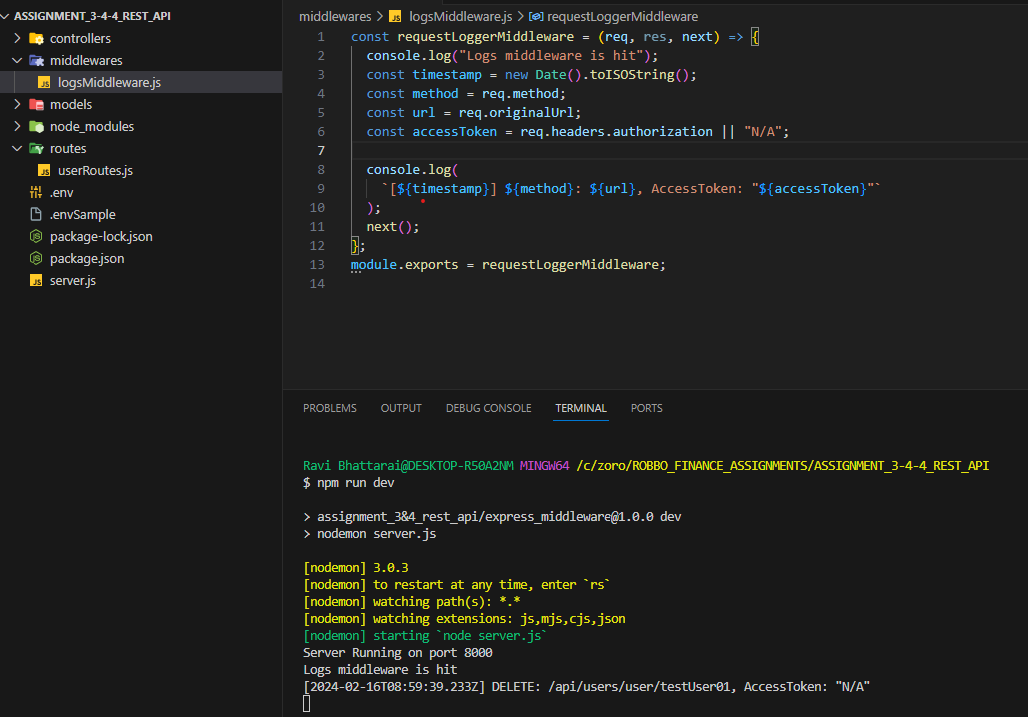
**Question 3: REST API with Node.js and Express, Question 4: Express Middleware Creation**

**And Question 5: MongoDB Data Aggregation are all included in the same project.**

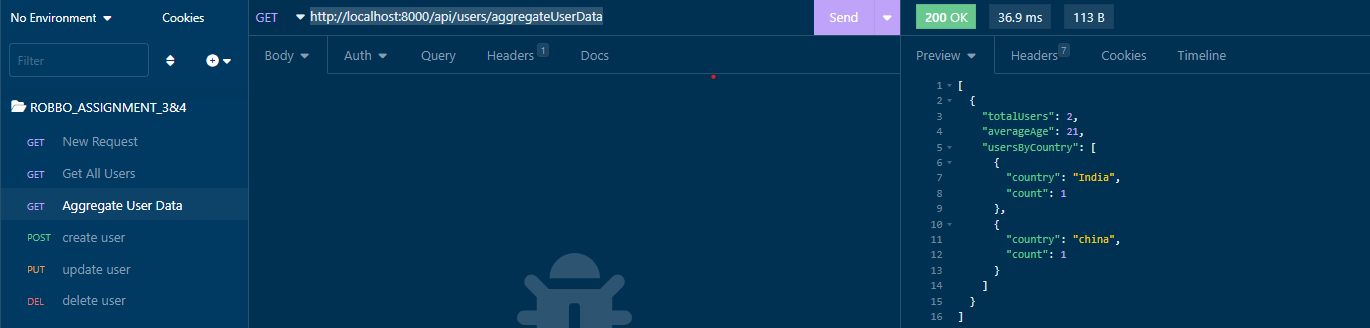
**Solution: Question 3: REST API with Node.js and Express**

* For Crud Operations please follow the below steps:
* GET/users 👍
  + Api url: <http://localhost:8000/api/users/users>
  + Output: 
* Create User: POST:
  + API URL: <http://localhost:8000/api/users/user>
  + 
* Update User:
  + API URL: PUT / [http://localhost:8000/api/users/user/](http://localhost:8000/api/users/user/testUser01):userId
  + Example: <http://localhost:8000/api/users/user/testUser01> , here testUser01 is userid, user Id will be automatically created by splitting the part before @ of email. Example: email is [testUser01@gmail.com](mailto:testUser01@gmail.com), then userId will be testUser01
  + 
* **Delete User**:
  + **API URL:** .DELETE / [http://localhost:8000/api/users/user/{userId](http://localhost:8000/api/users/user/%7BuserId)}
  + Example Api Url: <http://localhost:8000/api/users/user/testUser01>  
    After deletion of the user just a message is shown  
    
* Ensure passwords are not returned in the response.

**Solution: Question 3: Question 4: Express Middleware Creation**

* Hit any api url check the console of the terminal.
* The middleware code is written in source code’ middleware folder, logs Middleware.js
* 

**Solution: Question 5: MongoDB Data Aggregation**

* To check the aggregated data, hit the api url: GET / <http://localhost:8000/api/users/aggregateUserData>, 

* The code for the aggregation can be found in:
  + /controllers/userController.js