

Instructions:

- This exercise carries 50 points.
- The exercise contains Description of the problem, requirement, hint and expected output.
- Read the instruction clearly before practicing the exercises.
- The answers must be named after the naming convention mentioned in each exercise.
- Each exercise done by student is expected to meet the learning outcome mentioned in each exercise.
- All the exercises must be completed on time.
- You may discuss about the problem conceptually with the trainer or fellow students or from internet **however getting direct answers or copying code from others or from internet is strictly prohibited.**
- All the exercise files must be submitted to google classroom.
- All the exercises will be evaluated and points will be updated by the end of the bootcamp.
- Based on the points obtained by each student the overall ranking will be prepared.
- **All the documents (including this document) used in this bootcamp are sole properties of KIT and for internal purposes only. Must not be shared to anyone outside of this bootcamp.**

EXERCISE 01:**Description:****Setting up the db:**

- Create the database **my_tasklist** in the **Atlas or mlab** create the **user name as** : <your_name> and **password as: 1234** (Don't create the collection, it has to be created from the server side code).
- Copy the connection string (URI of the DB) after you have created the db (To be used later in express.js)

Setting up the project:

- Create a project in VSCode editor, Create package.json file and add the required dependencies **express, cors, body-parser, nodemon, mongodb driver dependencies**. (Research and write the use cases of each dependencies in the below mentioned sheet)

Setting up the middleware:

- Create index.js under server folder (server/index.js).
- Setup index.js as the default file for node and nodemon in the package.json file.
- Set up and listen to port 5000 in the index.js file.

Creating the API methods:

- Route the request for the URL: api/tasklist to tasklist.js from the middleware. (Research and write what is middle ware and write in the docx file mentioned in the requirements)
- In tasklist.js create an async loadTasksCollection() that connect with the database using String URL that you have copied after creating the database, create a collection mytasks and return.
- Create the router.get() that reads the mytasks collection and return all the tasks. (Routed for the url '/').
- Create the router.post() that inserts a new task to the mytasks collection and return the status code 201 if successfully inserted (Fields to be inserted: **task** and **dateCreated**). (Routed for the url '/').
- Create the router.delete() that deletes an existing task from the mytasks collection and return the status code 200 if successfully deleted. (Routed for the url '/<_id of the task to be deleted>').
- Use Postman (A software tool to send http requests – Very useful during real-time development) to test with the http request.

Requirements:

- Screenshots of the steps and results must be added to the word file **01_Mod_03_FE_to_BE_Research.docx** and to be submitted to the google classroom.
- The entire project must be uploaded to Git Hub and the URL of the project must be submitted in the Google classroom.

Expected Outcome:

- To be able to create a cloud database
- To be able to understand how to create APIs and middleware using Express.js.
- To be able to connect express with Mongo db.
- To be able to understand how to test with Postman.

ALERT

Make sure that you follow all instructions properly. Be aware that each character is important. Upload your solutions for all the exercises on the google classroom created and shared with you.