SIT323/SIT737- Cloud Native Application Development 4.1P: Building a simple calculator microservice

Overview

The aim of this task is to create and deploy a microservice offering basic calculator functionality to clients. This microservice should effectively handle incoming requests and execute fundamental arithmetic operations (e.g., addition, subtraction, multiplication, division) on provided input numbers. Utilize Node.js and Express for developing the microservice. is to design and implement a microservice that provides a simple calculator functionality to clients. The microservice should be able to handle incoming requests and perform basic arithmetic operations (e.g., addition, subtraction, multiplication, division) on the input numbers. You will be using Node.js and Express to build the microservice.

The required tools for doing this task are as follows:

- Git (https://github.com)
- Visual Studio Code (https://code.visualstudio.com/)
- Node.js (https://nodejs.org/en/download/)

Part I- Requirements

- The microservice should be able to accept requests for arithmetic operations in a standard format (e.g., REST API).
- The microservice should be able to handle errors and provide meaningful error messages to clients.

Part II- Instructions

Configure and Setup the Microservice *(You can follow the instruction with your Tutor in the class as well)*

1. Set up the development environment: Install Node.js and any necessary dependencies to build and run the microservice.
2. Create a new Node.js project: Create a new Node.js project and install the Express framework using npm.
3. Design the API endpoints: Decide on the API endpoints for the microservice. For this exercise, you will need four endpoints: one for addition, one for subtraction, one for multiplication, and one for division. Each endpoint should take two input parameters: num1 and num2.
4. Implement the API endpoints: Use Express to implement the API endpoints. Make sure to handle incoming requests and perform the appropriate arithmetic operation on the input parameters.
5. Handle errors: Implement error handling to provide meaningful error messages to clients. For example, if the input parameters are not valid numbers, the microservice should return an error message indicating that the parameters are not valid.

Submission Details

- Once you are done, push your code into your repo, giving the repository the following name sit323-2025-prac4p, ultimately this should read as https://github.com/username/ sit323-2025-prac4p. You can copy/paste the link of your public repo for your submission through the OnTrack;
- Ensure that detailed documentation is included with your code, offering step-by-step instructions that explain the process undertaken for this part. Failure to provide this documentation will result in an incomplete mark for the task.