Task 4.1P – Documentation

Repo link : Pasindufdo98/sit737-2025-prac4p

This project is a basic calculator microservice built using Node.js and Express. It includes four API endpoints to perform arithmetic operations: addition, subtraction, multiplication, and division. Additionally, the microservice uses the Winston logging library to track incoming requests and errors, which is useful for development and troubleshooting.

Step-by-Step Instructions

Microservice Development

1. Initialize a Node.js Project

- Open the terminal and create a new project folder. Then run the following commands:

  npm init -y

  npm install express

2. Create the Express Application

   Create a file called index.js and set up a basic Express server that listens on port 3040.

3. Design and Implement API Endpoints

   Four GET endpoints were created: /addition, /subtraction, /multiplication, /division

4. Add Input Validation and Error Handling

   The service checks for valid number inputs and handles division by zero errors with appropriate messages.

Adding Winston Logging

1. Install Winston

   Run npm install winston

2. Configure Winston Logger

   Logs errors to error.log, all info-level logs to combined.log, and console logs during development.

3. Log Request Information

   Logs operation type, input values, and errors.

Run the Application

Start the server with node index.js

Test the Endpoints

ex: Use browser to test the end points;

http://localhost:3040/addition?n1=10&n2=5

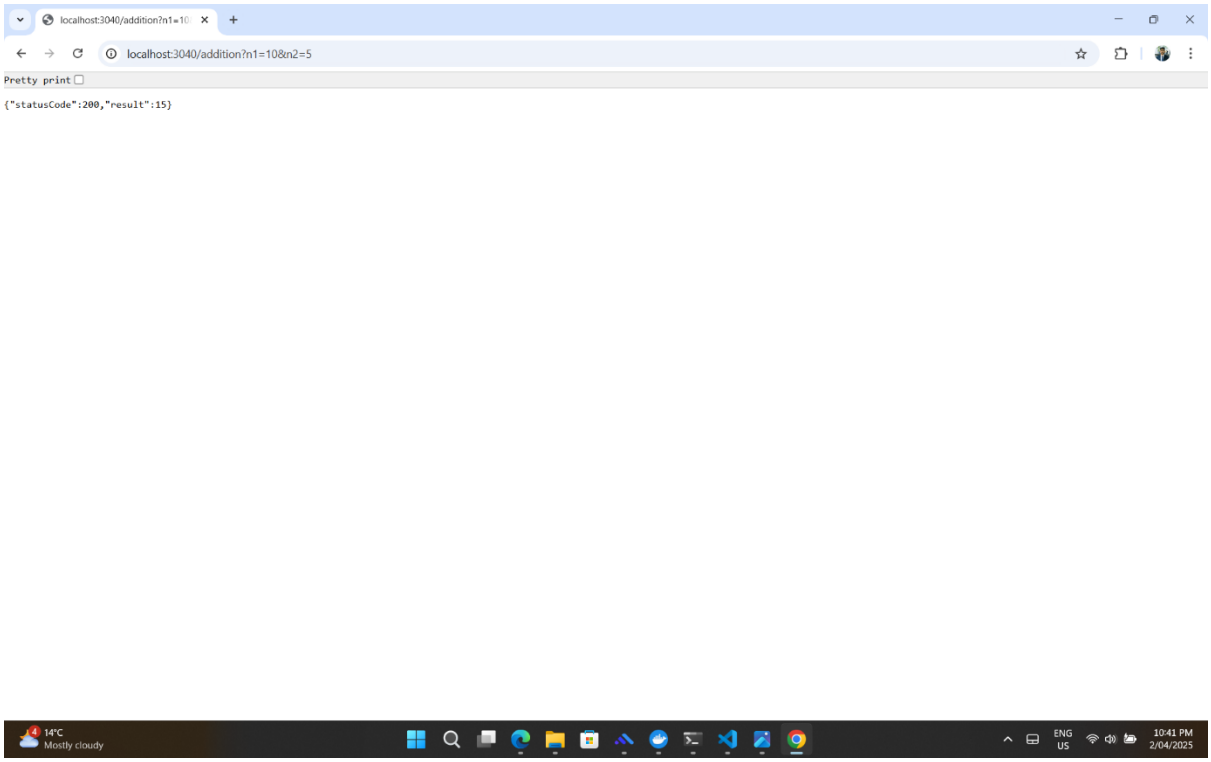
http://localhost:3040/substraction?n1=10&n2=3

http://localhost:3040/division?n1=10&n2=5

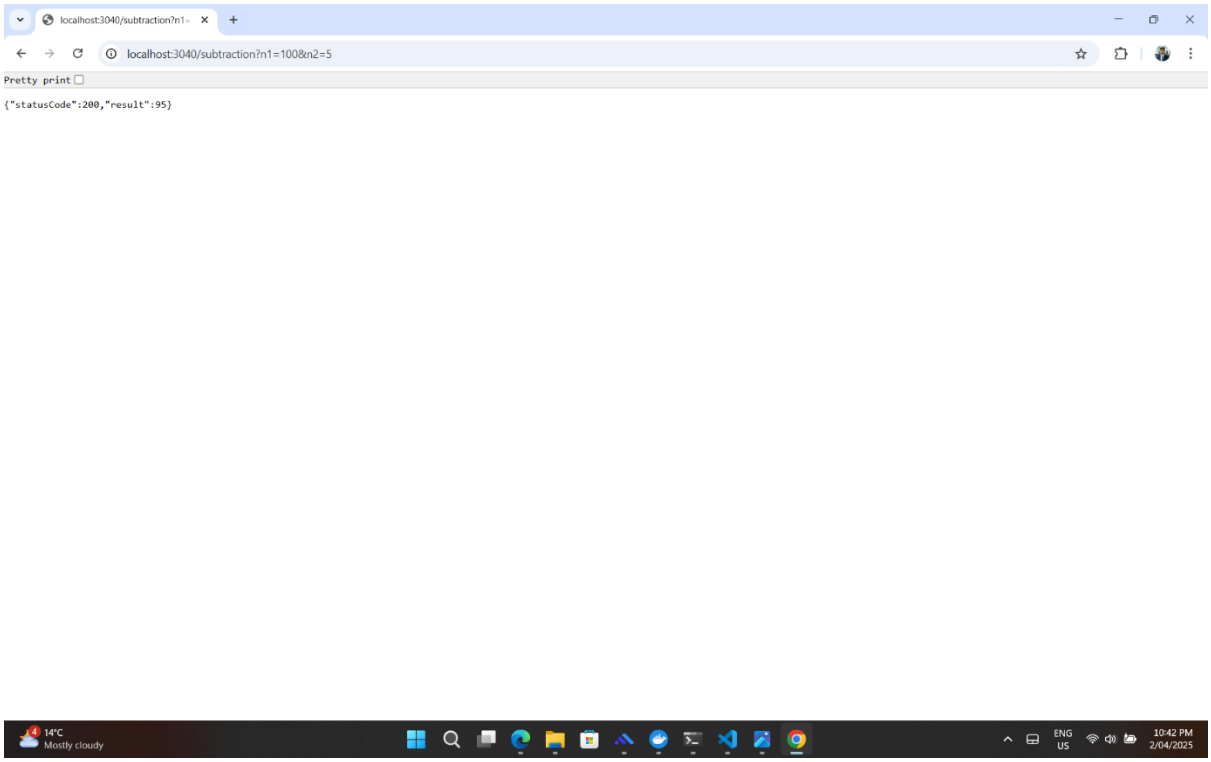
http://localhost:3040/multiplication?n1=100&n2=4

Screenshots

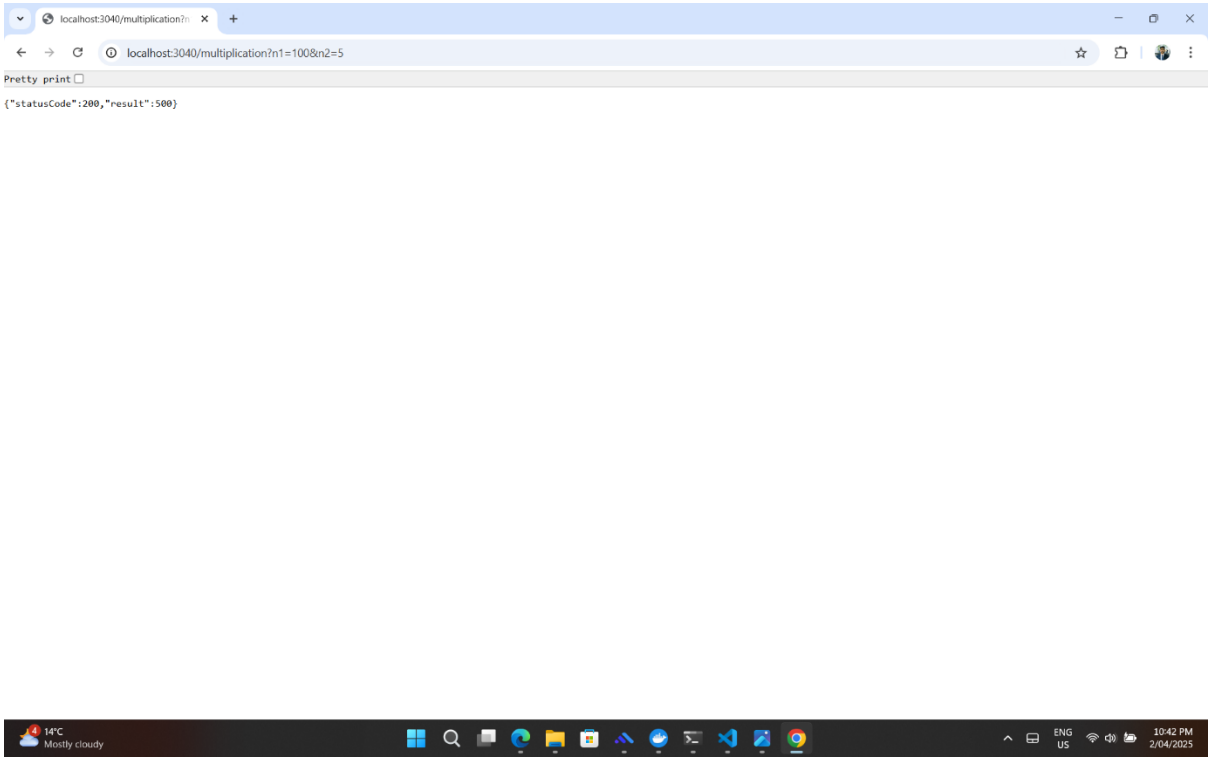
Addition



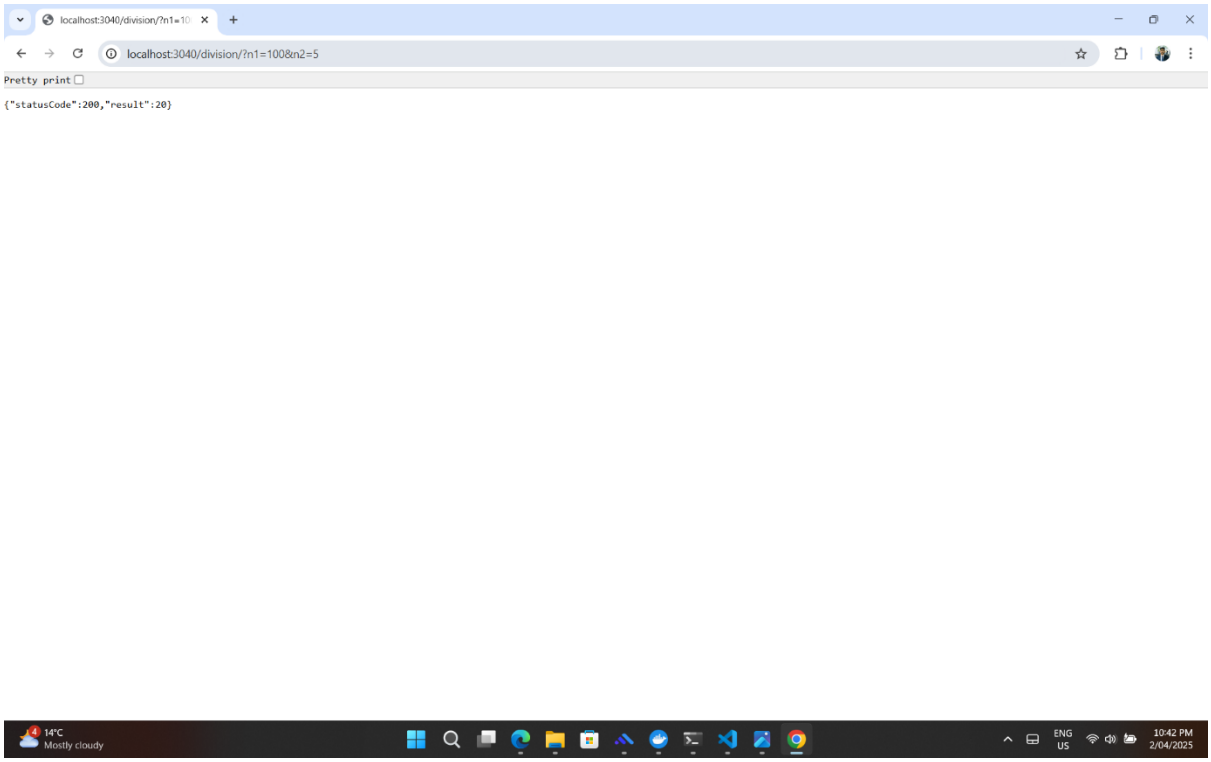
Subtraction



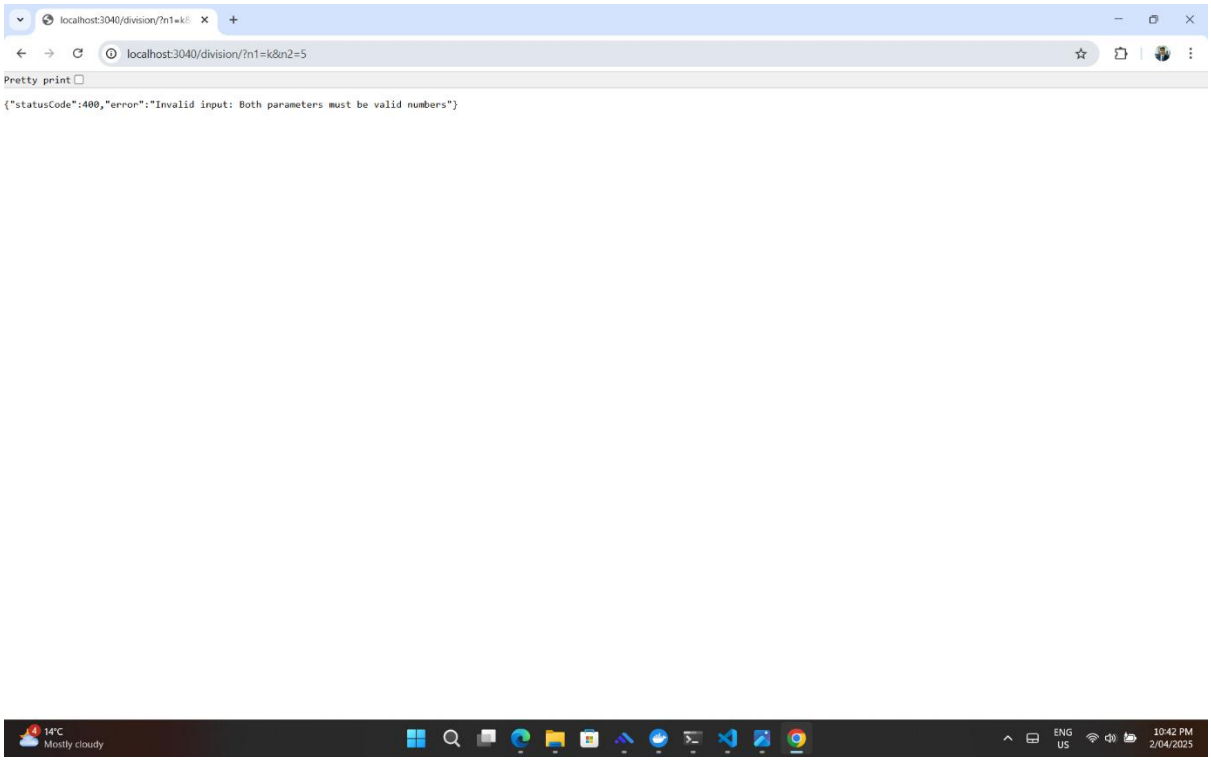
Multiplication

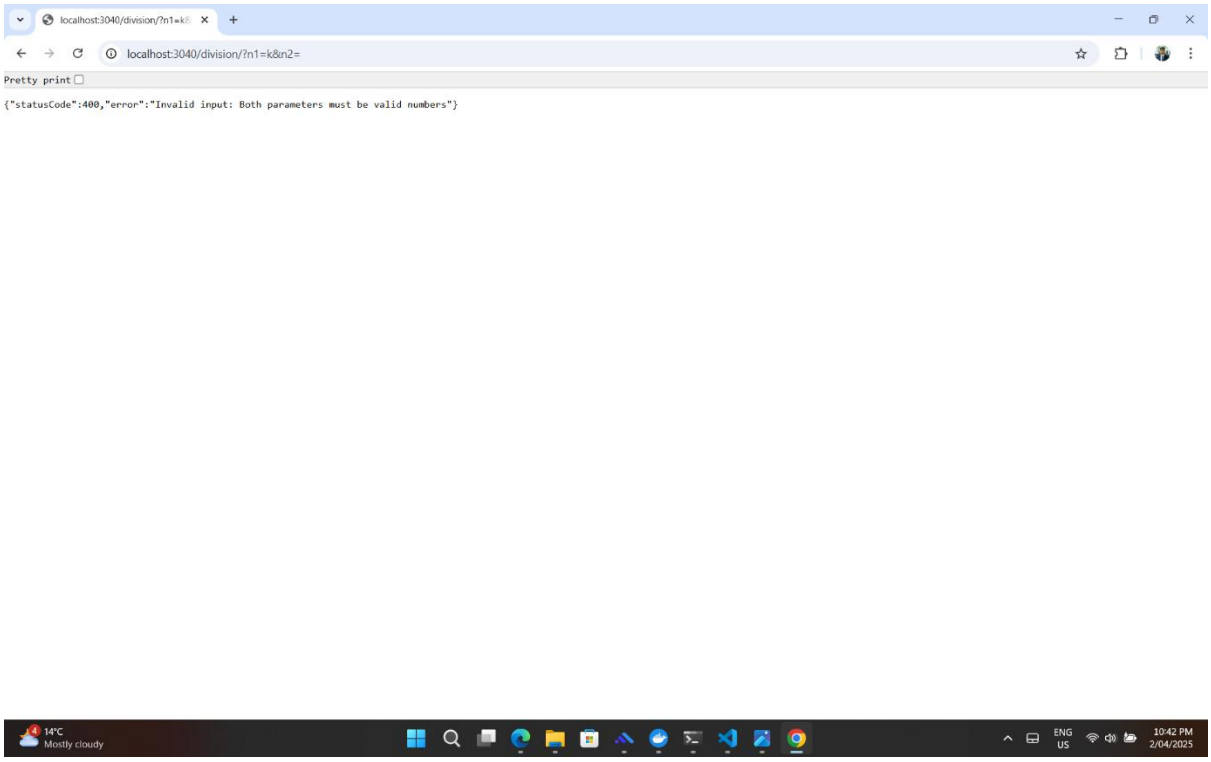


Division

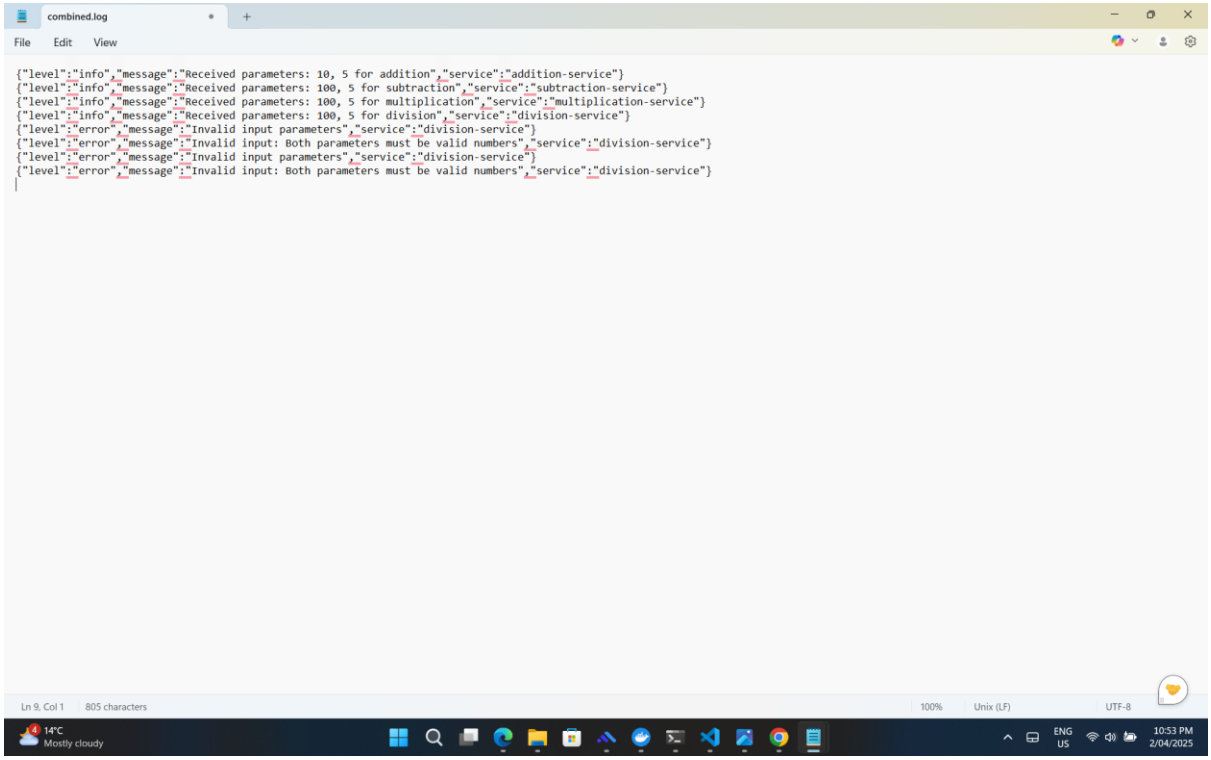


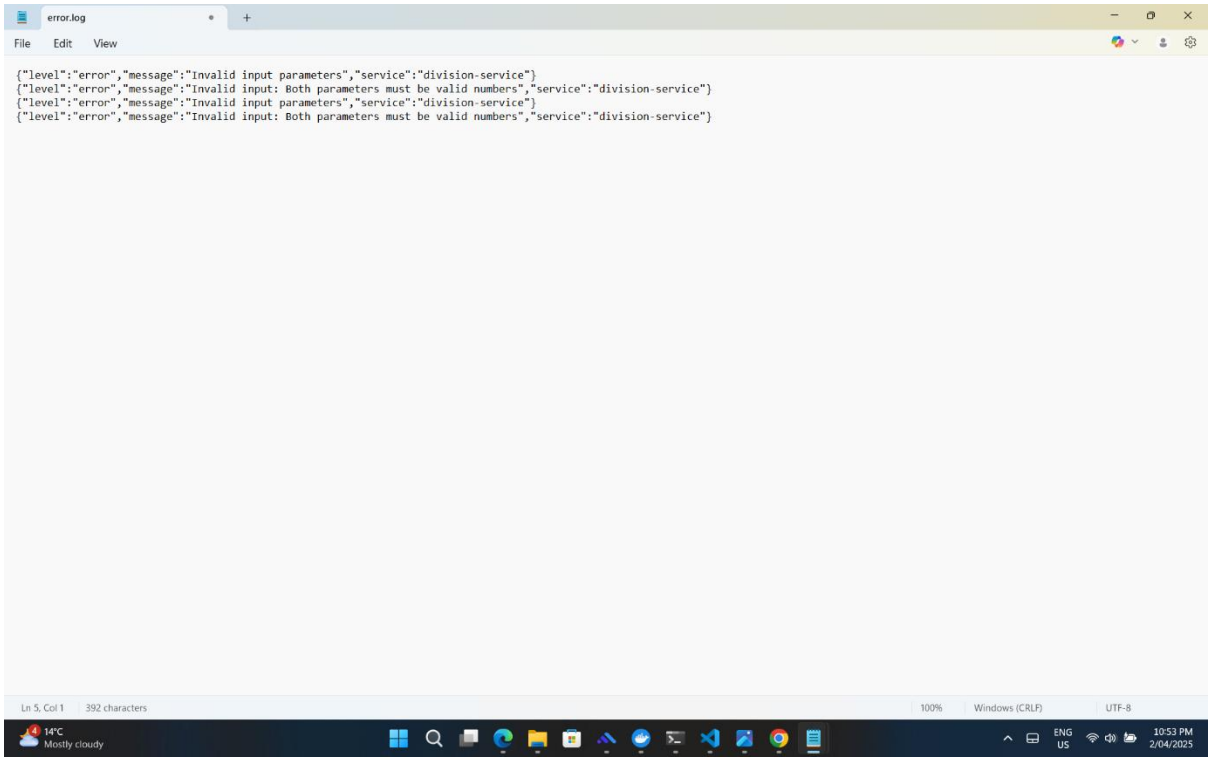
Invalid Parameters



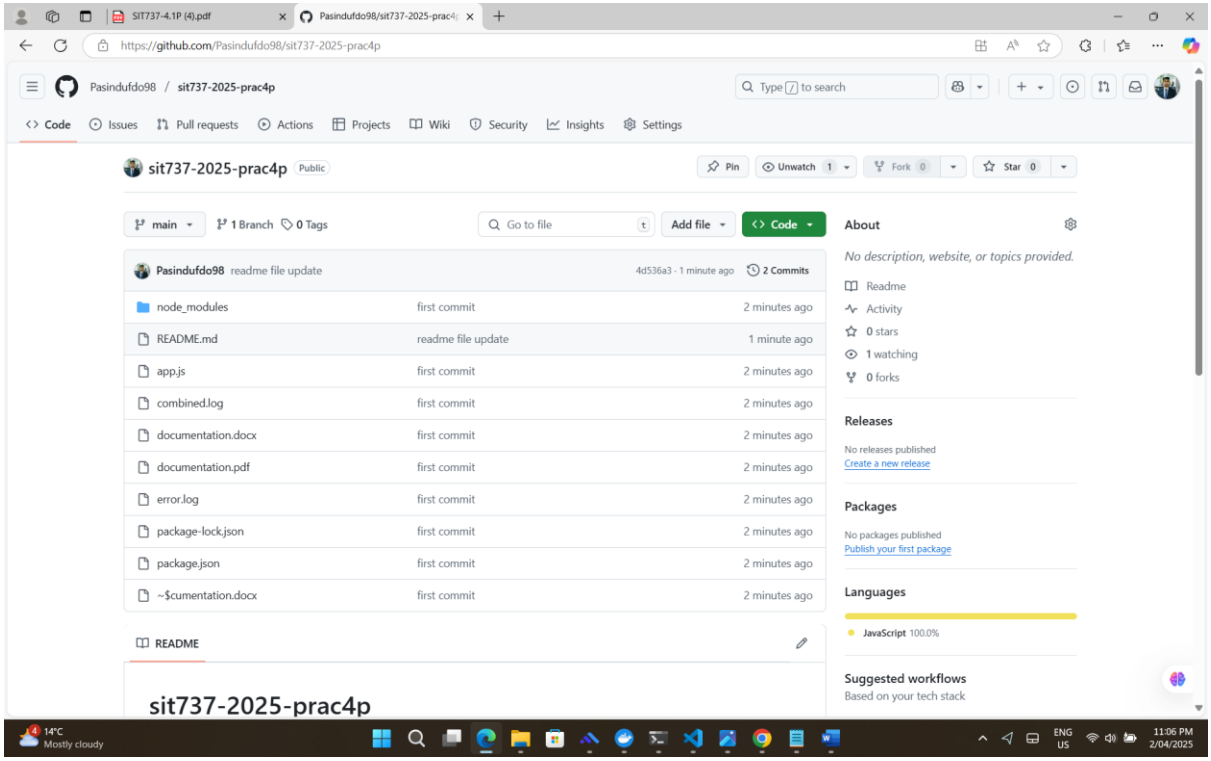


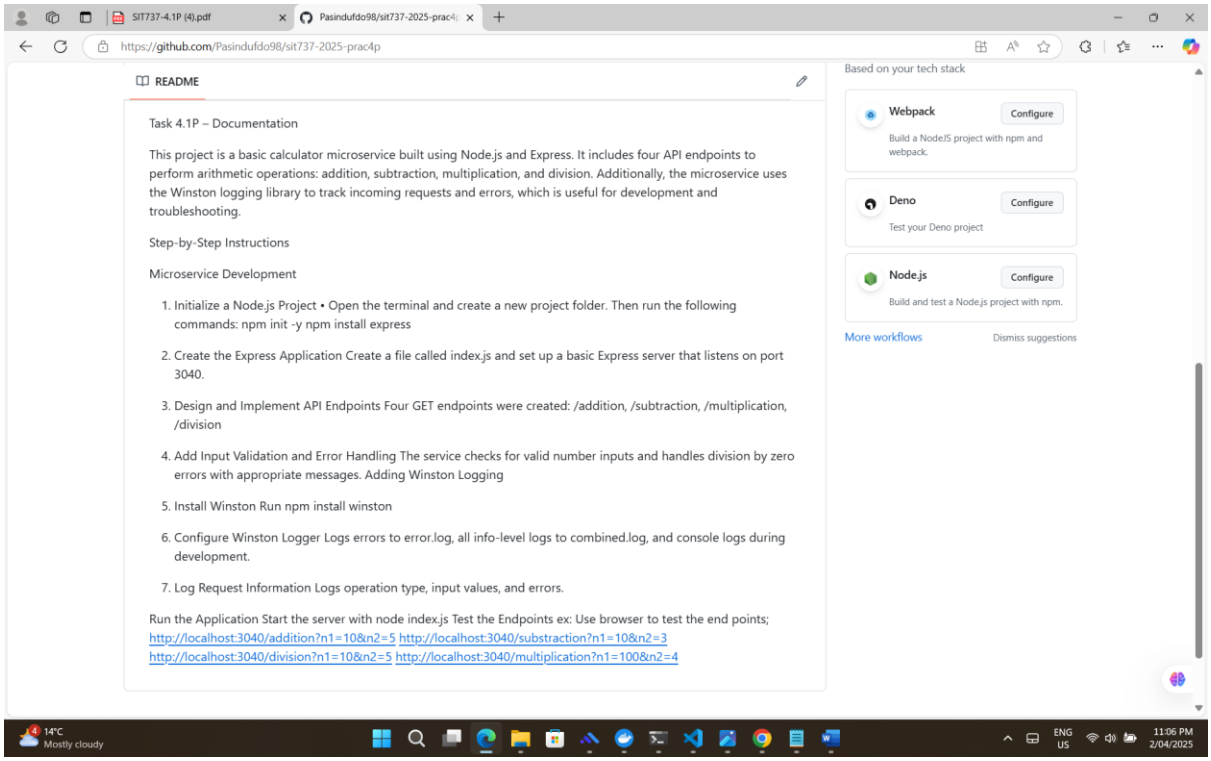
Winston Log files




error.log
File Edit View
{"level":"error","message":"Invalid input parameters","service":"division-service"}
{"level":"error","message":"Invalid input: Both parameters must be valid numbers","service":"division-service"}
{"level":"error","message":"Invalid input parameters","service":"division-service"}
{"level":"error","message":"Invalid input: Both parameters must be valid numbers","service":"division-service"}
Ln 5, Col 1 392 characters 100% Windows (CRLF) UTF-8
14°C Mostly cloudy
ENG US
10:53 PM 2/04/2025

GitHub Repo


https://github.com/Pasindufdo98/sit737-2025-prac4p
Pasindufdo98 / sit737-2025-prac4p
Code Issues Pull requests Actions Projects Wiki Security Insights Settings
sit737-2025-prac4p Public
Pin Unwatch 1 Fork 0 Star 0
main 1 Branch 0 Tags Go to file Add file Code
Pasindufdo98 readme file update 4d536a3 · 1 minute ago 2 Commits
node_modules first commit 2 minutes ago
README.md readme file update 1 minute ago
app.js first commit 2 minutes ago
combined.log first commit 2 minutes ago
documentation.docx first commit 2 minutes ago
documentation.pdf first commit 2 minutes ago
error.log first commit 2 minutes ago
package-lock.json first commit 2 minutes ago
package.json first commit 2 minutes ago
~$cumentation.docx first commit 2 minutes ago
README
sit737-2025-prac4p
About
No description, website, or topics provided.
Readme
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Languages
JavaScript 100.0%
Suggested workflows
Based on your tech stack
14°C Mostly cloudy
ENG US
11:06 PM 2/04/2025


SIT737-4.1P (4).pdf
Pasindufdo98/sit737-2025-prac4p
https://github.com/Pasindufdo98/sit737-2025-prac4p
README
Task 4.1P – Documentation
This project is a basic calculator microservice built using Node.js and Express. It includes four API endpoints to perform arithmetic operations: addition, subtraction, multiplication, and division. Additionally, the microservice uses the Winston logging library to track incoming requests and errors, which is useful for development and troubleshooting.
Step-by-Step Instructions
Microservice Development
1. Initialize a Node.js Project • Open the terminal and create a new project folder. Then run the following commands: npm init -y npm install express
2. Create the Express Application Create a file called index.js and set up a basic Express server that listens on port 3040.
3. Design and Implement API Endpoints Four GET endpoints were created: /addition, /subtraction, /multiplication, /division
4. Add Input Validation and Error Handling The service checks for valid number inputs and handles division by zero errors with appropriate messages. Adding Winston Logging
5. Install Winston Run npm install winston
6. Configure Winston Logger Logs errors to error.log, all info-level logs to combined.log, and console logs during development.
7. Log Request Information Logs operation type, input values, and errors.
Run the Application Start the server with node index.js Test the Endpoints ex: Use browser to test the end points;
http://localhost:3040/addition?n1=10&n2=5 http://localhost:3040/substraction?n1=10&n2=3
http://localhost:3040/division?n1=10&n2=5 http://localhost:3040/multiplication?n1=100&n2=4
Based on your tech stack
Webpack
Configure
Build a NodeJS project with npm and webpack.
Deno
Configure
Test your Deno project
Node.js
Configure
Build and test a Node.js project with npm.
More workflows
Dismiss suggestions
14°C
Mostly cloudy
ENG
US
11:06 PM
2/04/2025