Node.js Task: To-Do Application (Pure Node.js)

Objective

Create a small To-Do application using pure Node.js (without Express or any external framework). The goal is to understand how Node.js handles HTTP requests, file operations, and data persistence.

Task Instructions

1. Create a Node.js HTTP Server

Use the built-in http module to create a simple web server that listens on a specific port (e.g., 3000).

2. Data Storage

Store all to-do data in a file named todo-data.json. Use the fs module to read and write data to this file. The data must remain available even after the server restarts.

3. Add a New To-Do Task (POST)

Create an API endpoint to add a new task. Read the request body and append the new task to the file. Assign each task a unique ID.

4. View All To-Do Tasks (GET)

Create an endpoint to return all the existing to-do tasks from the file. Send the data as a proper HTTP response.

5. Edit an Existing To-Do Task (PUT)

Implement an endpoint that updates a task's content or status by its ID. Modify the corresponding record in the file.

6. Delete a To-Do Task (DELETE)

Implement an endpoint that removes a task by its ID. Update the file after deletion.

7. Persistence Requirement

Make sure all operations update the file so that the data is not lost when the server is restarted.

8. Optional Improvements (Bonus)

Add basic validation for invalid IDs or missing fields. Add an endpoint to mark all tasks as completed. Add timestamps for created and updated tasks.