Problem Statement: Task Manager REST API

Objective: Create a RESTful API in Go for managing tasks. The API should allow users to create, read, update, and delete tasks. Users should be able to register, log in, and access their own tasks only when authenticated.

Requirements:

1. **Authentication & Authorization**:
    - Implement user registration and login functionality.

Users should be able to create, read, update, and delete tasks only if authenticated.
Ensure that users can only access their own tasks.

2. **Task Management**:
    - Create endpoints for creating, reading, updating, and deleting tasks.

Each task should have at least a title, description, and status (e.g., "todo," "in progress," "done").
Implement proper validation for task data (e.g., required fields, valid status values).

3. **Database Integration**:
    - You can use your choice of database but you need to justify your decision in a readme document.

Middleware & Error Handling:
- Implement middleware for authentication.

Implement middleware for logging each incoming request.
Handle errors gracefully and provide meaningful error responses.

Concurrency:
- Create a route that allows users to mark multiple tasks as "done" concurrently.

Demonstrate the use of Goroutines and channels for concurrent processing.

Documentation:
- Include clear instructions on how to run the application and the API endpoints in a readme file.

Bonus Features (Optional):
- Implement pagination for listing tasks.

Add sorting and filtering options for task lists (e.g., by status, creation date).
Deploy the application to a cloud platform of your choice (e.g., AWS, Google Cloud).
Generate API documentation using a tool like Swagger or GoDoc.
Unit Test your code.

Deliverables:

- A well-structured Go application with clear code organization.

Documentation for the API, including how to register, log in, and use the endpoints.
A GitHub repository containing the code and documentation.
A brief write-up explaining your design choices and any additional features you implemented.

Evaluation:
“implement all possible best coding practices as you see fit. You shall be asked to justify/explain why given code style used and why some areas of best coding are omitted. Purpose of the exercise is to measure how best coder you are and how well code you can write. Purpose here is not to write working code but best code for the given implementation.”