Objective: Develop a backend for a task management system using Node.js, Express, MongoDB, and WebSocket for real-time updates.  
Requirements:

1. API Development with Express:

* Create RESTful endpoints to manage tasks: adding a new task, updating a task (completed/active status), deleting a task, and listing all tasks.
* Implement user authentication using JWT. Only authenticated users can perform operations on tasks.

1. Real-time Updates:

* Integrate WebSocket (or [Socket.IO](http://socket.io/)) to notify connected clients in real-time when a task is added, updated, or deleted.

1. MongoDB Integration:

* Store tasks and user information in MongoDB. Each task should have a title, description, status (completed/active), and an associated user.
* Ensure tasks can be retrieved in an efficient manner.

1. Error Handling and Data Validation:

* Properly handle errors and validate incoming data to ensure the integrity of the API.

1. Documentation:

* Provide a README file that includes setup instructions, API endpoint documentation, and an overview of the real-time update mechanism.

Evaluation Criteria:

* Efficiency and Logic: Code should be efficient, logical, and solve the problem effectively.
* API Design: RESTful practices, appropriate response status codes, and a logical endpoint structure.
* Real-time Functionality: Correct implementation of real-time updates for task changes.
* Database Usage: Effective use of MongoDB for storing and querying data.
* Security and Validation: Implementation of JWT-based authentication, proper data validation, and error handling.
* Documentation: Clear documentation on how to set up and use the API, including any necessary environment setup.

Optional Bonus:

* Implementing a simple rate-limiting mechanism to prevent abuse of the API.
* Using environment variables to configure the application (e.g., database connection settings).