Technical Test Subject: Real-Time Chat Application with Node.js and TypeScript

Objective

Develop a real-time chat application using Node.js and TypeScript that demonstrates your understanding of streams and events. The application should be well-documented and showcase your ability to write clean, maintainable code.

Requirements

1. **Basic Features:**
    - Users should be able to join the chat application with a unique username.
    - Users can send messages to the chat room, and all connected users should receive these messages in real-time.
    - Messages should include a timestamp and the sender's username.
2. **Streams:**
    - Implement a feature that logs all chat messages to a file using streams. Each message should be appended to the log file in real-time as they are sent.
    - Use streams to read from the log file and display the last 50 messages when a user joins the chat.
3. **Events:**
    - Utilize Node.js EventEmitter to handle user connections, disconnections, and message events.
    - Create custom events for user join, user leave, and message received.
4. **Documentation:**
    - Provide detailed comments and documentation for your code.

  - Include a README.md file that explains how to set up and run the application, along with a brief overview of the code structure and design decisions.

Technical Details

- **Technology Stack:**
  - Node.js
  - TypeScript
  - Any library of your choice for handling WebSockets (e.g., Socket.io)
  - Use built-in Node.js modules for file handling and event management
- **Project Structure:**
  - Follow best practices for project structure in a Node.js/TypeScript application.
  - Separate concerns using different files and directories (e.g., `src/`, `logs/`, `types/`).
- **Submission:**
  - Share the GitHub repository link containing your project.
  - Ensure your code is well-formatted and adheres to TypeScript best practices.

Bonus Points

- Deploy the application on a cloud platform (e.g., Heroku, AWS) and provide the live URL in the README.md.
- Add unit tests for critical parts of the application.

Evaluation Criteria

- Correctness and functionality: Does the application meet the requirements?
- Code quality: Is the code clean, maintainable, and well-documented?
- Use of streams and events: Are streams and events utilized effectively?

- Completeness of documentation: Is the README.md file clear and comprehensive?