Objective:

Develop a Node.js API that handles user registration, authentication, and basic game data using Express for routing, MySQL for user data, MongoDB for game data, and RabbitMQ for event processing.

Tasks:

1. **User Registration and Authentication**

   Create a MySQL database schema for user data that includes fields for username, email, password, and any additional data you think is necessary.

   - Implement a registration endpoint that allows users to sign up by providing their username, email, and password. The password should be securely hashed before storing it in the database.
   - Implement an authentication mechanism that allows users to log in and receive a JSON Web Token (JWT) for subsequent API requests.

2. **Game Data API**

   Create a MongoDB collection for storing game data (e.g., player statistics, game results, etc.). Implement endpoints to:

   - Create a new game entry with relevant data.
   - Retrieve game data for a specific user.
   - Update game data for a specific user.
   - Delete a game entry.

3. **RabbitMQ Event Processing**

   - Set up a RabbitMQ instance for event processing.
   - Implement an event publisher that sends a message to RabbitMQ whenever a user registers.
   - Implement an event subscriber that listens for these events and logs them in a file.

Additional Guidelines:

- Use appropriate middleware for request validation, error handling, and JWT token verification.
- Follow best practices for code organization, such as separating routes, controllers, and services.
- Provide a README file with instructions on setting up the development environment, including database and RabbitMQ configuration.
- Encourage the use of modern JavaScript/Node.js practices and coding standards.
- Not mandatory: Ensure proper API documentation (e.g., using Swagger or similar tools).

Assessment:

Assess the candidate's coding skills, understanding of Node.js, Express, MySQL, MongoDB, and RabbitMQ, as well as their ability to structure and document their code.