**Overview:**

In this adjusted project, the inclusion of real-time analytics remains an optional feature. Your primary focus will still be on developing a robust blogging platform using Node.js, Express, and either MongoDB or PostgreSQL, with containerization (Docker preferred). However, if you choose to implement the optional real-time analytics feature, you will leverage Socket.IO to add real-time updates and notifications to the platform. This project aims to demonstrate your versatility in developing both traditional and cutting-edge web applications.

**Core Requirements:**

1. **User Service**:  
   - Implement user registration and login functionality.  
   - Allow users to update their profiles.

2. **Blog Service**:  
   - Enable users to create, read, update, and delete blog posts.

3. **Database**:  
   - Choose between MongoDB or PostgreSQL based on the needs of the services and your preference.

4. **Containerization**:  
   - Containerize both microservices using Docker or any other service (Docker is preferred), ensuring they can communicate as necessary.

5. **Basic Security Measures**:  
   - Implement JWT for securing API endpoints.  
   - Sanitize inputs to mitigate common web vulnerabilities.

6. **Documentation**:  
   - Provide basic documentation for your APIs, including available endpoints and how to authenticate requests.

**Optional Feature: Real-Time Analytics Service**

If you decide to implement this feature:

- **Integration of Socket.IO**: Enhance both the User and Blog Services with Socket.IO for real-time analytics.  
- **Interest-Based Notifications**: Implement a system for users to specify their interests and receive notifications for new blog posts matching those interests in real time.  
- **Admin Monitoring**: Provide an admin dashboard that uses Socket.IO to display real-time statistics about the platform.

**Deliverables:**

- Two microservices (User and Blog) with basic CRUD operations.  
- Containers for each microservice.  
- Basic API documentation.  
- If implemented, documentation and source code for the real-time analytics feature.

**Evaluation Criteria:**

- Completeness of the microservices according to the specified requirements.  
- Quality and efficiency of the Dockerized setup.  
- Clarity and usefulness of the provided documentation.  
- If applicable, creativity and effectiveness of the real-time analytics solution.

**Timeline:**

- Days 1-2: Setup and initial development of the User Service.  
- Days 3-4: Completion of the Blog Service.  
- Day 5: Integration and testing of both services.  
- Day 6: Containerization of the services.  
- Day 7: Documentation and final touches.

**Notes:**

- Focus on completing the core requirements first. The optional real-time analytics feature is meant to challenge you creatively without detracting from the main objectives.  
- Use environment variables for configurations where possible to simplify deployment.  
- Consider using a tool like Swagger for API documentation to make it more accessible and interactive.

This adjusted task allows for flexibility, enabling you to prioritize the essential features while optionally exploring advanced real-time capabilities, showcasing your comprehensive skills in full-stack development.