Home Test Assignment - Professional Development Test

Time Limit: 3.5 hours

Overview

Build a complete full-stack application with authentication, database integration, and monitoring capabilities.

Note: You are permitted to use any tools, resources, and assistance during this assignment.

Important: After submission, you should expect follow-up questions about your implementation. You must be able to explain all aspects of your solution in detail, including architecture decisions, how each component works, database design, Docker configuration, logging implementation, and any AI-assisted code.

Part 1: Simple Development

Requirements:

1. **Create a simple website that connects to a database using:**
   - **Backend**: Node.js
   - **Frontend**: React or basic HTML (your choice)
   - **API**: RESTful API with Node.js
   - **Database**: TiDB (required)
2. **Implement Login Screen:**
   - User authentication interface
   - Username/email and password fields
   - Form validation
   - **Note: Keep the client side very basic - UI design is not important**
3. **User Token Management:**
   - User tokens stored in database
   - Tokens sent as HTTP headers for authenticated requests

Part 2: DevOps Implementation

Requirements:

1. **Dockerize the Services:**
   - Create Docker containers for client and API services
   - Write Dockerfiles for each service
2. **Database Setup:**
   - Use **TiDB** database
   - Configure TiDB in Docker environment
3. **Message Queue Integration:**
   - Implement **Apache Kafka** as message broker
   - Set up Kafka brokers in Docker environment
4. **Database Initialization:**
   - When Docker loads, automatically:
     - Import database tables structure
     - Create a default user with password

Part 3: Monitoring & Logging (SRE Implementation)

Requirements:

1. **User Activity Logging:**
   - Every time a user logs in, write a log entry in JSON format to console
   - Use **log4js** (required)
   - Log format should include: timestamp, user ID, action, IP address
2. **Database Change Monitoring:**
   - Every update/insert/delete operation in the database should be logged
   - Implement using **Change Data Capture (CDC) with TiDB**
   - Log database changes to console in structured format
3. **Real-time Data Processing:**
   - Write a **consumer application in Node.js** that:
     - Consumes database change messages from Kafka
     - Processes these changes and writes them to console
     - Maintains the same structured logging format

Technology Stack:

- **Frontend**: React or basic HTML
- **Backend**: Node.js with Express.js
- **Database**: TiDB
- **Message Queue**: Apache Kafka
- **Containerization**: Docker & Docker Compose
- **Logging**: log4js (required)

Submission Requirements:

1. Complete source code in a Git repository
2. Docker configuration files (docker-compose.yml, Dockerfiles)
3. Database schema and seed files
4. README.md with setup and running instructions
5. **The entire project must be runnable with a single command: either one script or `docker-compose up`**

Good luck with your assignment!