Backend Setup (Node.js and Hasura)

1. **Node.js Setup**:
    - Install Node.js if you haven't already. You can download it from the official website.
    - Create a new directory for your project.
    - Initialize a Node.js project using npm init.
2. **Hasura Setup**:
    - Install Hasura GraphQL Engine using Docker or directly from the Hasura website.
    - Set up your database (PostgreSQL) and configure Hasura to connect to it.
    - Define your data model (tables, relationships, etc.) using Hasura's console.
3. **Business Logic**:
    - Create custom Hasura Actions using Node.js. These actions will handle business logic like deposits and withdrawals.
    - Define the logic for deposit and withdrawal operations in your Node.js server.
    - Use Hasura Actions to call your Node.js functions.

Frontend (HTML/CSS/JavaScript)

1. **HTML/CSS Setup**:
    - Create an HTML file for your frontend.
    - Design a simple user interface with forms for deposit and withdrawal.
    - Style your UI using CSS.
2. **JavaScript (Frontend Logic)**:
    - Use JavaScript to handle form submissions.
    - Send requests to your Hasura GraphQL API for deposit and withdrawal operations.
    - Display relevant messages to the user based on the API responses.

Security Measures

1. **Authentication and Authorization**:
    - Implement user authentication (e.g., JWT tokens) to secure your endpoints.
    - Define user roles (admin, regular user) in Hasura and restrict access to certain actions based on roles.
2. **Validation and Sanitization**:
    - Validate user input (e.g., amount for deposit/withdrawal) to prevent malicious data.
    - Sanitize inputs to avoid SQL injection.

Deliverables

1. **Source Code**:
    - Share the complete source code for both the backend (Node.js) and frontend (HTML/CSS/JavaScript).
2. **Setup Instructions**:
    - Document how to set up the project locally (including Node.js, Hasura, and database setup).
    - Explain how to run the backend and frontend together.
3. **API Documentation**:
    - Provide detailed documentation for your Hasura GraphQL API endpoints.
    - Include sample requests and responses.
4. **Design Decisions and Assumptions**:
    - Explain any design choices you made during development.
    - Document any assumptions you've made about the project requirements.