Tech Assignment: Return and Exchange Management System

Objective:

The goal of this assignment is to implement a backend service for managing **Return and Exchange** requests for an e-commerce platform. The solution should use the following tech stack:

- **Backend**: Node.js with Express.js
- **ORM**: Sequelize
- **Database**: Schema code present in the end of this doc

Figma Link for frontend flow:
https://www.figma.com/design/l02oUGIudAEzHLOGNIm2bY/lz-return-%26-exchange?node-id=0-1&p=f&t=7cptJ8qaecupWCRc-0

Please Note, the above link has checkout and tracking flow as well. However, you should focus on return & exchange.

Business Logic

1. **Request Creation**:
    - Customers can raise a return/exchange request.
    - The request is reflected to both the seller and the admin, alongwith customer app & web.

2. **A. Seller Action Return** :
    - The seller has 24 hours to either accept or reject the request.
    - The seller also specifies whether the return is required or not.
    - If the seller does not respond within 24 hours, the admin takes over.

   B. Seller Action Exchange :

    - The seller has 24 hours to either accept or reject the request.
    - The seller provides a reason for their decision.
    - If the seller does not respond within 24 hours, the admin takes over.

3. **Admin Review**:
    - Admin reviews the request after the seller updates or after 24 hours.
    - Admin's decision (approve/reject) is final.
4. **A. Return Flow**:
    - **Approved + Return Required**: A reverse pickup is scheduled, and a refund is issued after the pickup is completed.
    - **Approved + Return Not Required**: Refund is issued directly.
    - **Rejected**: A regret response is sent to the customer with the status updated.

    B. Exchange Flow:

    - **Approved**: A reverse pickup is scheduled, and a tracking link is updated.
    - **Rejected**: A regret response is sent to the customer with the reason and status updated.

5. **Communication**:
    - Update the request status throughout the process.
    - Admin’s final decision is communicated to both the seller and the customer via email.

Requirements

1. **Routes**
    - CRUD operations for handling return and exchange requests based on the business logics.
    - Routes for customer, seller, and admin actions.
2. **Database Models**
    - Implement Sequelize models for managing requests.
    - Schema attached in annexure for reference
3. **Business Logic**
    - As outlined above
4. **Error Handling & Validations**
    - Validate all inputs (e.g., reason for rejection, customer details).
    - Handle edge cases (e.g., missing data, invalid status transitions).
5. **Documentation**
    - API documentation using tools Postman (optional but encouraged).
6. **Testing**
    - Basic unit tests for core business logic.

Deliverables

1. Source code in a GitHub repository with ReadME
2. Screen recording of the deployment on Loom
3. Postman API Collection
4. Deployed Link