Readme.md

Transport Management System (TMS)

A backend service built using Spring Boot and PostgreSQL to manage logistics operations including loads, transporters, bidding, and booking workflows with real-world constraints and concurrency handling.

📌 Overview

This system simulates a real Transport Management workflow where shippers post loads, transporters submit bids, and bookings are created based on accepted bids. The service enforces business rules such as capacity validation, load status transitions, multi-truck allocation, and optimistic locking for concurrency control.

🏗️ Architecture

```
Controller → DTO → Service (Business Rules) → Repository → Entity → Postgr
eSQL
```

Key Practices Used:

- Layered Architecture
- DTO-based request/response mapping
- Global Exception Handling (`@ControllerAdvice`)
- Optimistic Locking (`@Version`)
- Pagination and Filtering Support

🛠️ Tech Stack

Component	Technology
Language	Java 17

Component	Technology
Framework	Spring Boot 3.2+
ORM	Spring Data JPA
Database	PostgreSQL
Testing	JUnit + MockMVC
Documentation	Postman
Build Tool	Maven

📂 Project Structure

```
src/main/java/com/tms/
  ├── controller
  ├── service
  ├── repository
  ├── entity
  ├── dto
  └── exception
```

🚚 Core Domain Entities

- **Load**
- **Transporter**
- **Bid**
- **Booking**

Each follows constraints such as status transitions, validation rules, and relational mapping.

⚙️ Setup Instructions

1️⃣ Clone Repository

```
git clone https://github.com/dhopesarvesh/tms_backend.git
cd tms-backend
```

2️⃣ Configure Database

`application.yml`:

```
spring:
  datasource:
    url: jdbc:postgresql://localhost:5432/tms_db
    username: yourusername
    password: yourpassword
```

3️⃣ Run Application

```
mvn clean install
mvn spring-boot:run
```

🚀 API Endpoints

Postman APIS / COLLECTION available at

> https://web.postman.co/workspace/My-Workspace~95a57b2c-d215-43b2-82b9-c05fd57b1317/collection/41647254-0f83dae2-6f11-42a1-be82-10d60da2f077?action=share&source=copy-link&creator=41647254

Main Modules

Module	Endpoints
Load APIs	Create, List, Fetch, Cancel, Best Bids
Transporter APIs	Register, Update trucks, Fetch details
Bid APIs	Submit bid, View bids, Reject bid
Booking APIs	Create booking, Cancel booking, View booking

🧠 Business Rules Implemented

- Transporter bidding limited by available trucks
- Prevent bidding on cancelled/closed loads
- Multi-truck allocation support
- Load status auto-transitions based on actions
- @Version ensures conflict resolution in concurrent bookings
- Best-bid score logic: `score = (1 / proposedRate) * 0.7 + (rating / 5) * 0.3`

🗄️ Database Schema