🚙 Vehicle Transport System

Aim:

To develop a console-based and GUI-based Java application for managing a vehicle transport system.

To enable Admins and Drivers to manage vehicles, schedules, maintenance, and logs.

To support the tracking of vehicle usage, operational costs, and due dates for documentation like PUC and insurance.

Technologies Used:

- Java SE Programming
- Classes & Objects, Inheritance, Polymorphism
- Composition / Association / Aggregation
- Java Collections Framework
- Serialization & File I/O
- Exception Handling
- Java Swing for GUI

Duration:

- Console-Based Implementation: 30 Hours

- GUI Development with Swing: 8 Hours

Description:

- The Vehicle Transport System is used by a transport organization to:
- Maintain a list of vehicles and drivers.
- Assign travel schedules.

Record travel, fuel, and accident logs.

Track maintenance, PUC, and insurance details.

Generate monthly expense reports based on logs.

There are two user roles:

🛠 Admin Register Vehicles and Drivers

Plan Travel Schedules

Add Maintenance Records (PUC/Insurance)

Generate Monthly Expense Reports

Validate Upcoming Due Dates (for PUC/Insurance)

👷 Driver Add Travel Logs (date, distance)

Add Fuel Entries (liters, cost)

Report Accidents (description, date)

System Features:

👤 Customer (Driver) Management Add/Edit/Delete driver details

View all registered drivers

🚙 Vehicle Management Add/Edit/Delete vehicles

Track Insurance and PUC expiry

View maintenance history

📅 Travel Scheduling Schedule travel plans for drivers and vehicles

Avoid schedule conflicts

📓 Logging Activities Travel entries: Date, kilometers

Fuel entries: Liters and cost

Accident logs: Description and date

Maintenance records: Type, date

📊 Reporting Generate monthly expense reports (fuel, maintenance, accident)

Display due PUC/Insurance items

Track daily/monthly business operations

Functionality Overview:

🔧 **Vehicle & Driver Management**

- Add, view, and manage vehicle and driver information
- Ensure no duplicate entries using unique IDs

📅 **Travel Scheduling**

- Assign vehicles and drivers to travel schedules
- Prevent schedule conflicts

🗺️ **Travel Logging**

- Drivers log travel details such as kilometers covered

⛽ Fuel Entry

- Drivers log fuel refills with quantity and cost

🚨 Accident Reporting

- Drivers record incidents with descriptions and dates

🛠️ Maintenance Tracking

- Admin logs repair, maintenance, PUC, and insurance dates
- Upcoming expiry alerts

📊 Monthly Expense Reporting

- Admin generates reports summarizing fuel, maintenance, and accident expenses

---

System Design Highlights:

- Data is stored using appropriate collections (`HashMap`, `LinkedList`, `HashSet`)
- File I/O and serialization used to persist data
- Each `Vehicle` is linked to maintenance, travel, fuel, and accident records
- `ExpenseReport` aggregates data monthly for analysis

Database Entities:

- **Driver(DriverID, Name, LicenseNo, ContactInfo)**
- **Vehicle(VehicleID, RegNo, Type, Capacity, InsuranceDetails, InsuranceExpiryDate, PUCDate, PUCExpiryDate)**
- **Schedule(ScheduleID, Date, From, To, VehicleID, DriverID)**
- **TravelEntry(EntryID, Date, Km)**
- **FuelEntry(FuelID, Date, Liters, Cost)**
- **Accident(AccID, Date, Description)**
- **Maintenance(MaintID, Type, Date)**
- **ExpenseReport(ReportID, Month, TotalFuel, TotalMaintenance, TotalAccidentCost)**
