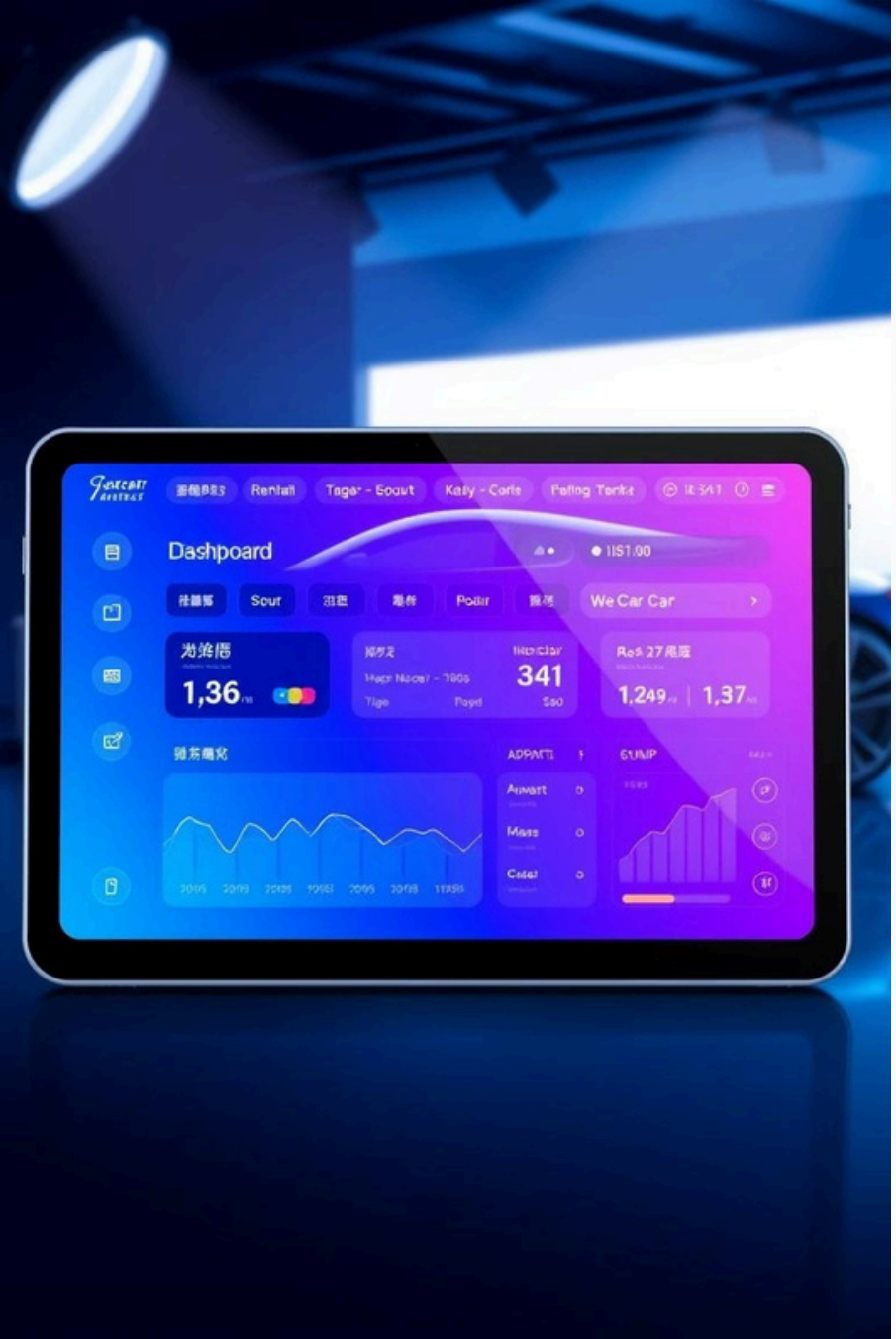
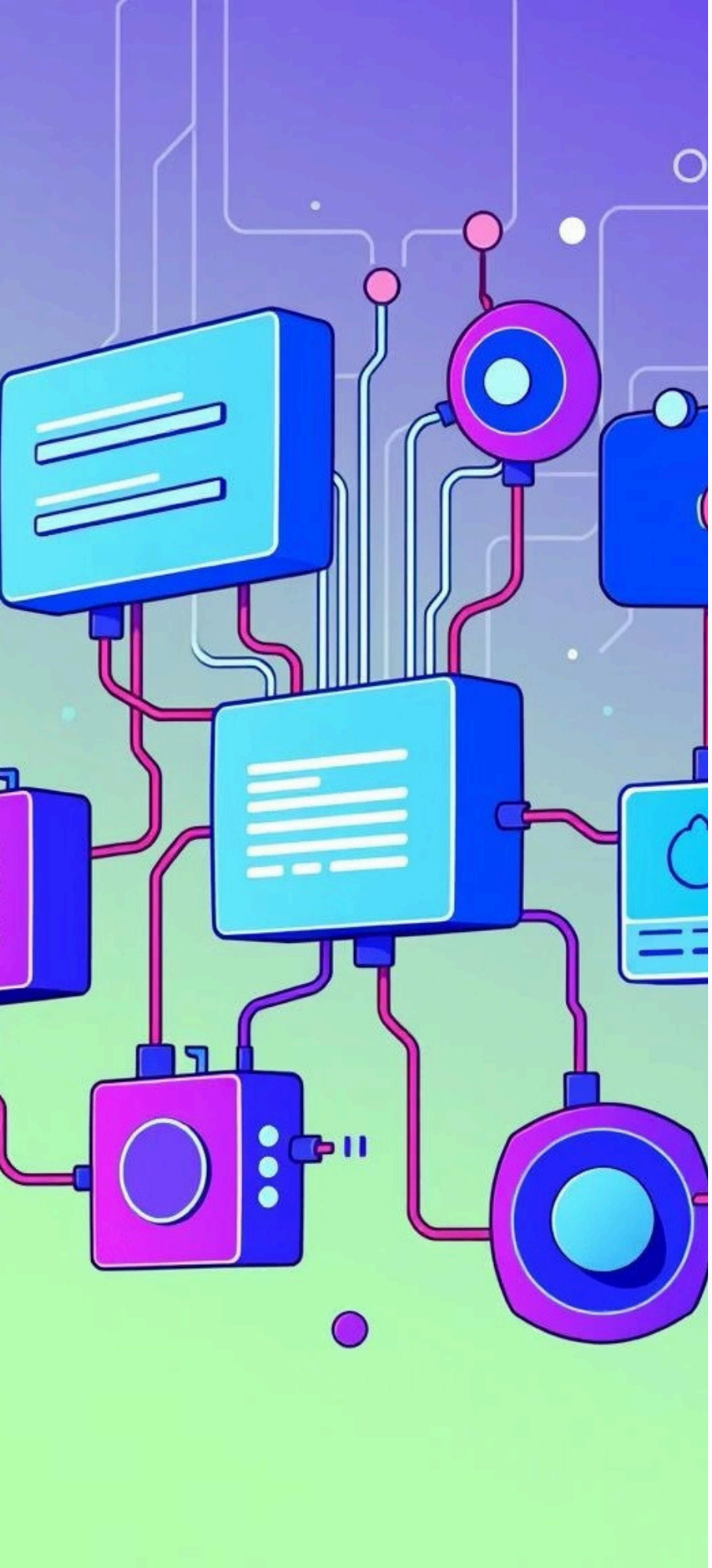


CAR RENTAL SYSTEM

structured design techniques. This kind of project helps rental companies streamline operations such as user registration, car booking, rental management, billing, and reporting while providing a user-friendly interface for customers and administrators.



WhyObject-Oriented Programming for Car Rentals?

Real-WorldMapping

Real-world entities like **Cars, Customers, and Rentals** map naturally to classes and objects, simplifying system conceptualisation.

Modular & Reusable Code

OOPprinciplesenablethecreation of **modular, reusable, and maintainable code**, fostering efficient development cycles.

Scalability & Flexibility

The architecturesupports **scalability**, making it easy to integrate new vehicle types, pricing models, or advanced features without extensive refactoring.

Industry Relevance

Languagessuch as**Java, C++, and Python** are widely used in Indian software projects, demonstrating the practical application of OOP concepts.

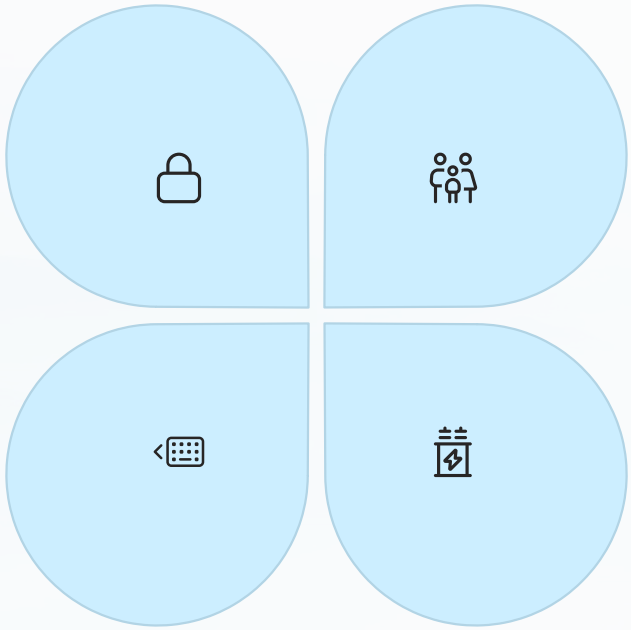
Core OOP Principles Applied

Encapsulation

Hiding internal data of classeslike Car and Customer, exposing only necessary methods to ensure data integrity and secur it y.

Abstraction

Defining interfaces for critical processes like payment processing or reservation handling, effectively hiding complex implementation details from the user.



Inheritance

Creating specialised vehicle classes (e.g., Sedan, SUV) that inherit common attributes and behaviours from a base Car class, promoting code reuse.

Polymorphism

Utilising method overriding to calculate rental fees differently based on the specific vehicle type, allowing for flexible pricing strategies.

Key System Modules & Their Responsibilities

1

Admin Module

Manages the entire fleet,including addingand removing vehicles, and overseeing all booking operations efficiently.

2

Customer Module

Allowscustomerstoregister, search for available cars, and make reservations seamlessly.

3

Vehicle Management

Tracks real-timecar availability, monitors maintenance schedules, and manages vehicle status updates.

4

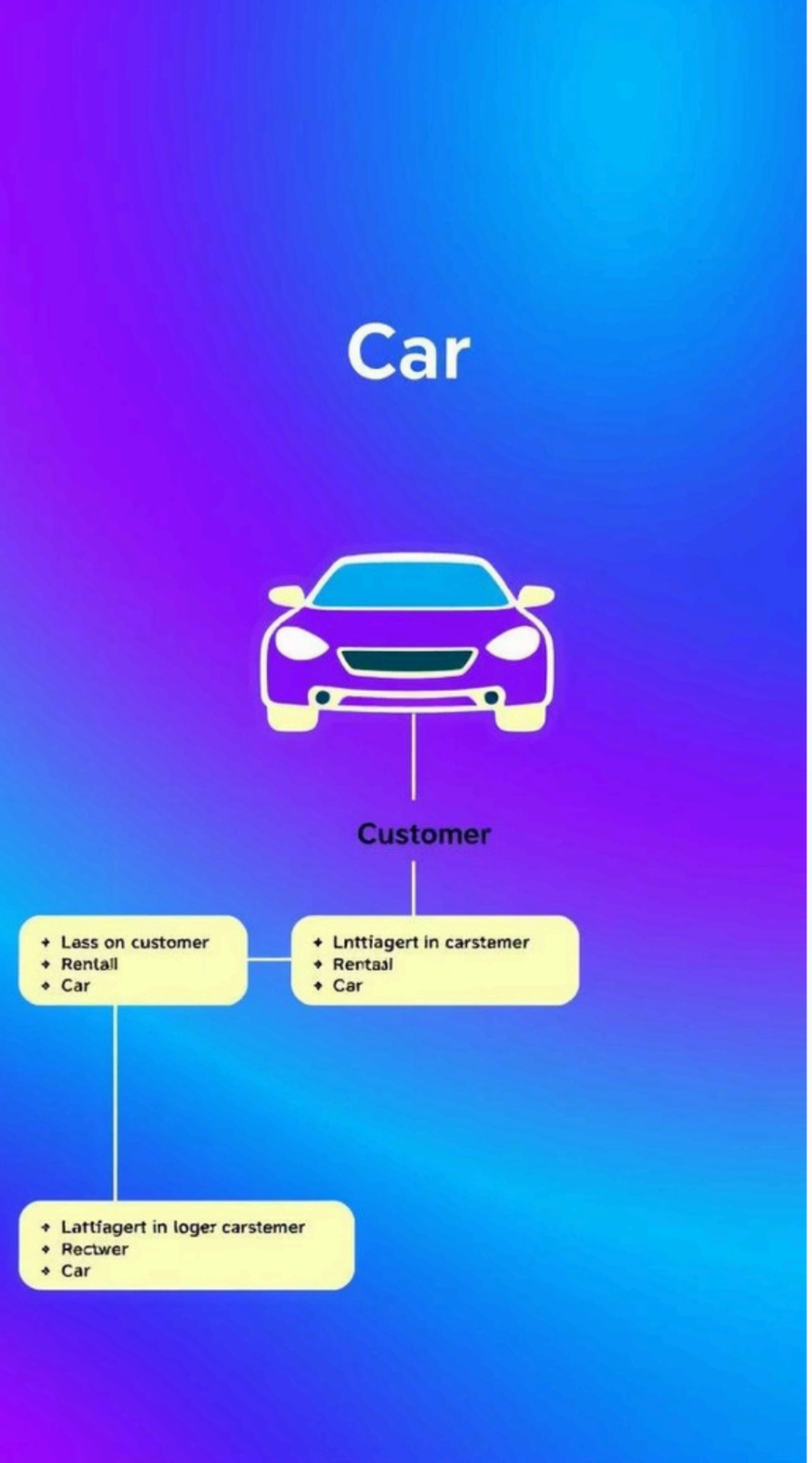
Rental & Billing

Calculates rentalfeesaccurately, processes payments securely, and generates detailed invoices for every transaction.

5

User Management

Handles user authentication and authorisation, ensuring secure access and personalised experiences.



Class Design Overview

1

Car Class

- Attributes: **carId, brand, model, pricePerDay, availability**
- Methods: **rent(), returnCar(), calculatePrice()**

2

Customer Class

- Attributes: **customerId, name, contact**
- Methods: **register(), viewRentalHistory()**

3

Rental Class

Links **CarandCustomer** objects, storing crucial rental details such as **duration and current status**.

4

CarRentalSystem Class

Manages**collections ofcars, customers, and rentals**, coordinating all core system operations for smooth functioning.

Example: Polymorphism in Rental Pricing

Base Car Class

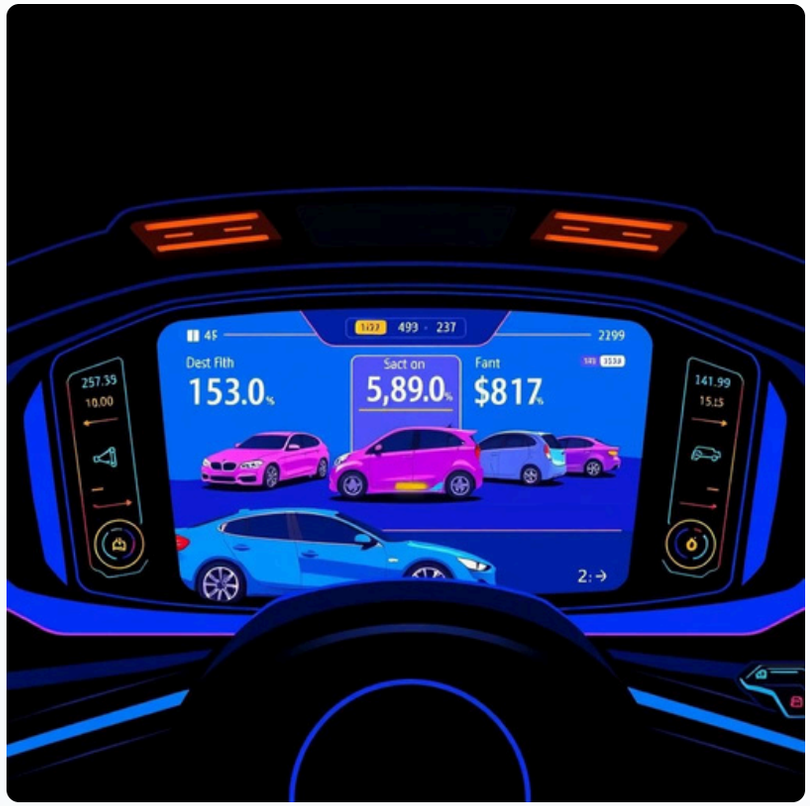
Definesastandard**calculatePrice(days)** method, establishing the fundamental pricing logic for all vehicles.

SUV Subclass

Overrides thebase method to incorporate **premium charges**, reflecting the higher value or specialised features of SUVs.

Sedan Subclass

Applies**discountsfor long-term rentals**, providing incentives for extended bookings and enhancing customer value.



This polymorphic approach allows for **flexible and dynamic pricing strategies** without altering the core system architecture.

Data Persistence & User Interaction

Data Storage

Utilise **Java serialization or simple text files** to store customer and rental data, ensuring basic persistence for system operations.

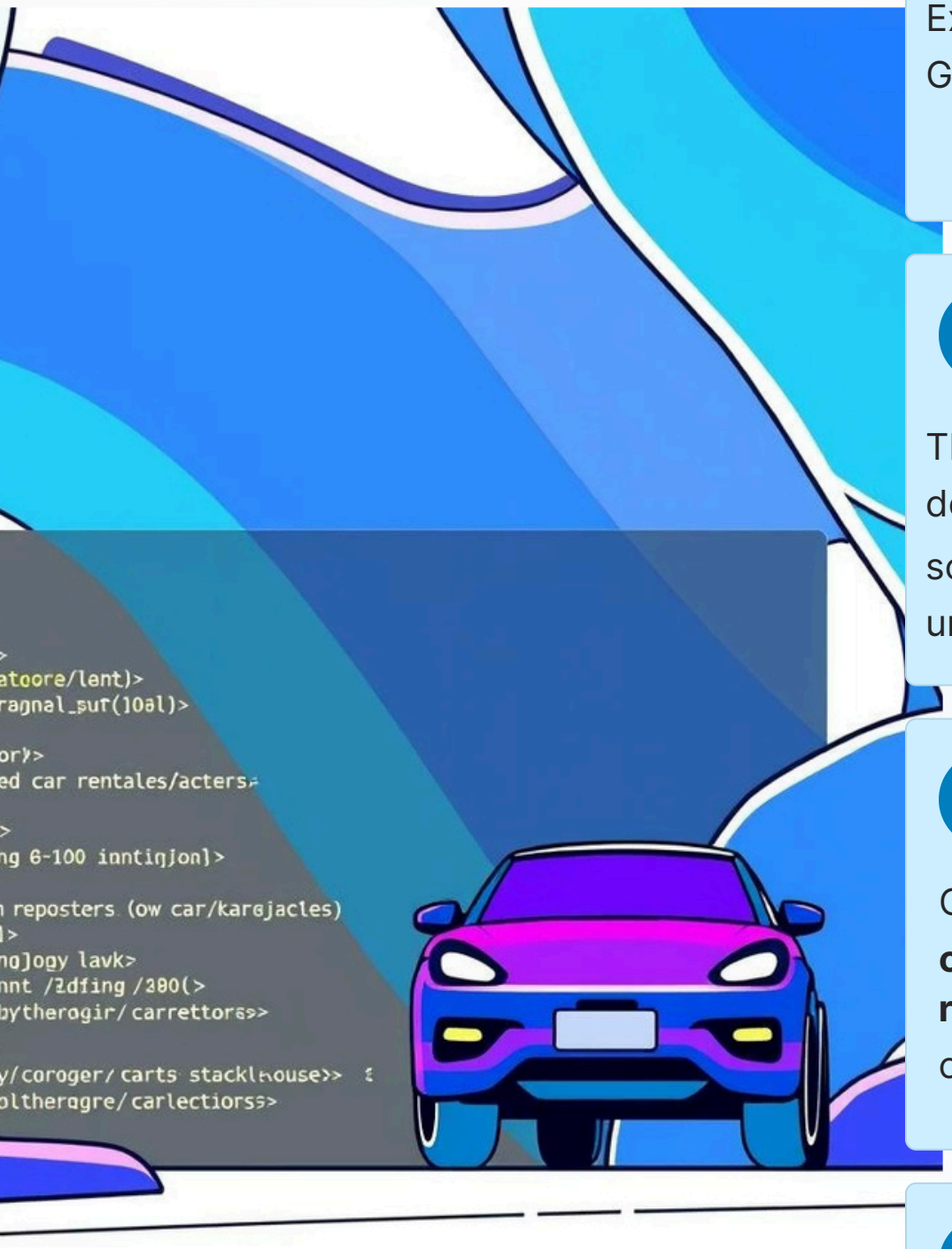
User Interface Options

Implement a **Command-Line Interface (CLI)** for quick interaction or a **GUI using Java Swing** for a more user-friendly experience.

Robust Error Handling

Integratecomprehensive**error handling mechanisms** to ensure system robustness, especially during crucial booking and car return processes.

Real-World Project Snapshot



GitHub Projects

Examples like**somnath-choudhury/car-rental-system** on GitHub showcase practical applications of these principles.

Modular Java Code

These projectstypicallyfeature **modular Java code** developed with
sound OOP design principles, making them easy to understand and extend.

Core Features

Commonfunctionalities include **car management, customer registration, and rental processing**, demonstrating a complete system.

User Experience

Theyoften provide**CLI menus** for a seamless and intuitive user
experience, ideal for rapid prototyping and deployment.

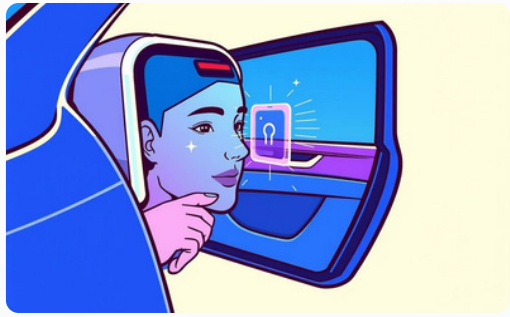
Future Enhancements with AI & Advanced Tech

AI-Driven Dynamic Pricing

Implement pricing models that adjust rates based on **real-time demand and individual user profiles**, optimising revenue.

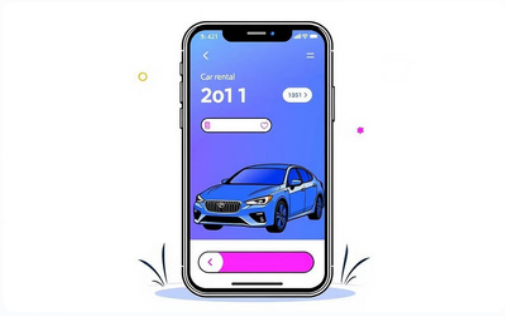
Fleet Maintenance Prediction

Leverage Altopredict vehicle maintenance needs, **reducing downtime and operational costs** by proactive servicing.



Contactless Rentals

Integrate**voice andface recognition** for secure, convenient, and entirely contactless car pickup and return processes.



Mobile App Integration

Developa dedicated mobile application for **on-the-go bookings, real-time vehicle tracking, and customer support**.

Conclusion: Building Robust Car Rental Systems with OOP

> As a result, the system reduces manual effort, enhances customer satisfaction, and improves business productivity

The Car Rental System AOODP project provides a streamlined, user-friendly, and automated solution for vehicle rental management using object-oriented design principles. By replacing manual and paper-based processes with computerized modules for booking, billing, inventory, and customer management, the system increases operational efficiency and accuracy while saving time and labor for both users and administrators

Thank You

TEAM MEMBERS:
D.GURU SAI TEJ REDDY-RA2411056010008