



Name of the School: School of Computer Science Engineering  
Course Code: R1UC201C Course Name: JAVA PROGRAMMING

# Car Rental System

Team Members :-

Submitted to :-

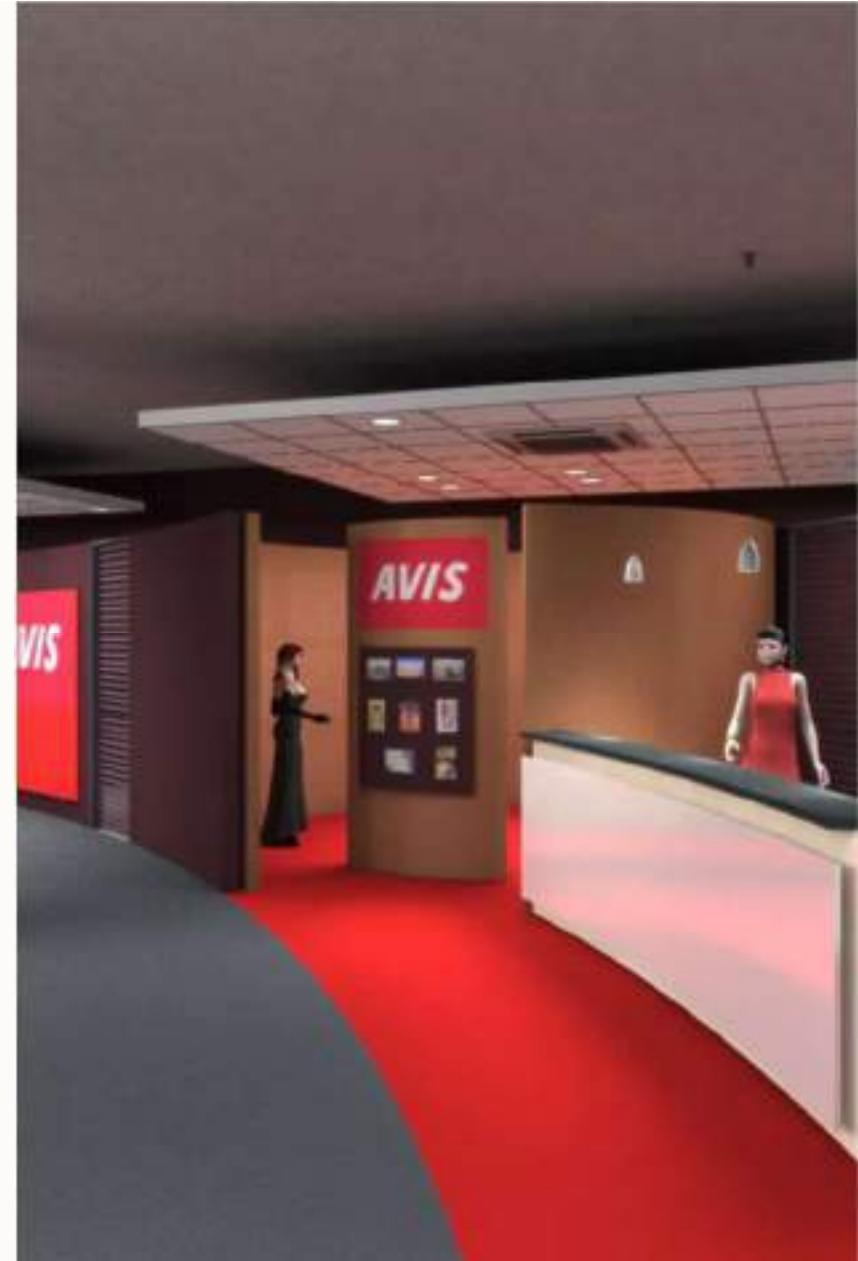
Programme Name: B.Tech (CSE)

# Car Rental System: Project Overview

This college project builds a simple yet functional car rental system.

Technologies used are Java, GUI, and MySQL database integration.

The system supports distinct Admin and User roles for management and rental.



```
CarRentalSystem/
├── src/
│   ├── model/
│   │   ├── User.java
│   │   ├── Car.java
│   │   └── Rental.java
│   └── dao/
│       ├── UserDao.java
│       ├── CarDAO.java
│       ├── RentalDAO.java
│       └── DBConnection.java
└── view/
    ├── LoginFrame.java
    ├── AdminDashboard.java
    ├── UserDashboard.java
    ├── CarManagementFrame.java
    ├── UserManagementFrame.java
    ├── RentalManagementFrame.java
    └── AvailableCarsFrame.java
└── Main.java
```

# System Architecture



## View Layer

GUI interface for user interaction using Java Swing/FX.



## Model Layer

Classes like Car, Customer, and Rental represent core entities.



## Data Access Layer

DAO classes handle database operations via MySQL connections.

# Database Design (MySQL)

## Car Table

- car\_id (INT, PK)
- model, make (VARCHAR)
- year (INT)
- price\_per\_day (DECIMAL)
- availability (BOOLEAN)

## Customer Table

- customer\_id (INT, PK)
- first\_name, last\_name (VARCHAR)
- email, phone (VARCHAR)

## Rental Table

- rental\_id (INT, PK)
- car\_id, customer\_id (INT, FK)
- start\_date, end\_date (DATE)
- total\_price (DECIMAL)

# Admin Module: Add Car



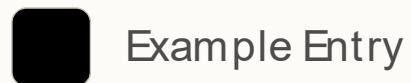
Car Details Form

Admins input Model, Make, Year, Price, and Availability.



DAO Method

`addCar(Car car)` in CarDAO  
saves new cars to database.



2023 Toyota Camry rented at \$45 per day.

The screenshot shows two windows of a software application. The top window is titled "Manage Cars" and displays a table of car rentals. The bottom window is titled "Car Information" and contains a form for adding a new car.

**Manage Cars**

carId	brand	model	status	costPerDay	outId	lastRental	Renter's Name
1	Toyota	Avalon	booked	450	4	2021-06-11T14...	Rajiv Bayota
2	Lamborghini	Urus	booked	500	2	2021-06-11T14...	Mallika Koirala
3	Nissan	GT-R	booked	550	10	2021-06-11T22...	John Cena
4	Chevrolet	Corvette	booked	550	1	2021-06-11T14...	Pablo Pascual
5	Audi	TTRS	available	530			
6	Jasper	Jean	available	839			
7	Ford	Raptor	available	381		2021-06-11T22...	
8	Jeep	Gladiator	available	400		2021-06-11T23...	

**Car Information**

Form fields:

- Brand:
- Cost Per Day:
- Model:
- Status:
- Search:  Car ID:
- Buttons: Save, Delete, Update, Mark As Available, Back

# Admin Module: Manage Customers



## Customer List

View, add, update, or delete customer records easily.



## DAO Operations

CRUD methods in CustomerDAO manage customer data efficiently.



## Key Methods

`addCustomer`, `updateCustomer`, and `deleteCustomer` used.

CRUD Actions		
	Add CRUD	Edit CRUD
<a href="#">Add CRUD to this table</a>		
	Read CRUD	Edit CRUD
	Edit CRUD	Delete CRUD
	Remove CRUD	Edit CRUD
	Edit CRUD	Delete CRUD
	Remove CRUD	Edit CRUD
	Add CRUD	Edit CRUD
	Read CRUD	Edit CRUD
	Edit CRUD	Delete CRUD
	Remove CRUD	Edit CRUD
<a href="#">Add CRUD to this table</a>		
<a href="#">Add CRUD to this table</a>		
	Add CRUD	Edit CRUD
	Read CRUD	Edit CRUD
	Edit CRUD	Delete CRUD
	Remove CRUD	Edit CRUD
<a href="#">Add CRUD to this table</a>		
	Add CRUD	Edit CRUD
	Read CRUD	Edit CRUD
	Edit CRUD	Delete CRUD
	Remove CRUD	Edit CRUD

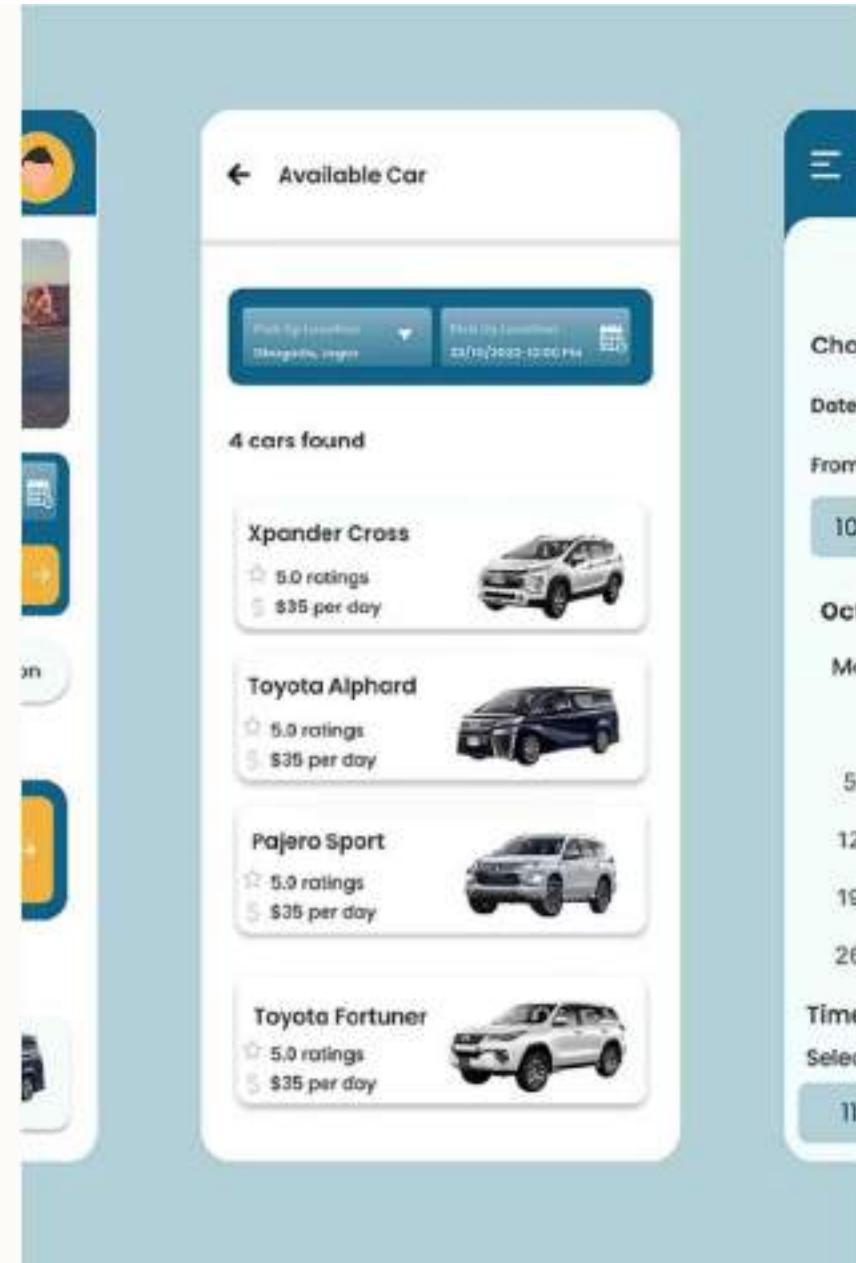
# User Module: Car Rental

## Car Search

Users filter cars by model, make, or rental dates.

## Availability Display

GUI table shows only cars free during chosen dates.





# User Module: Booking Process

## Car Selection

Choose a car and input rental start and end dates.

## Price Calculation

Total price computed based on rental duration and daily rate.

## Confirmation

Booking confirmation and payment processing on the spot.

# DAO Classes Implementation

## CarDAO

- getAllCars()
- getAvailableCars(Date start, Date end)
- addCar(Car car)
- updateCar(Car car)
- deleteCar(int carId)

## CustomerDAO

- getCustomer(int id)
- addCustomer(Customer customer)
- updateCustomer(Customer customer)
- deleteCustomer(int id)

## RentalDAO

- createRental(Rental rental)
- getRental(int rentalId)

# GUI Implementation (Java Swing/FX)



## Technology Choice

Built with Java Swing or JavaFX for responsive GUI.



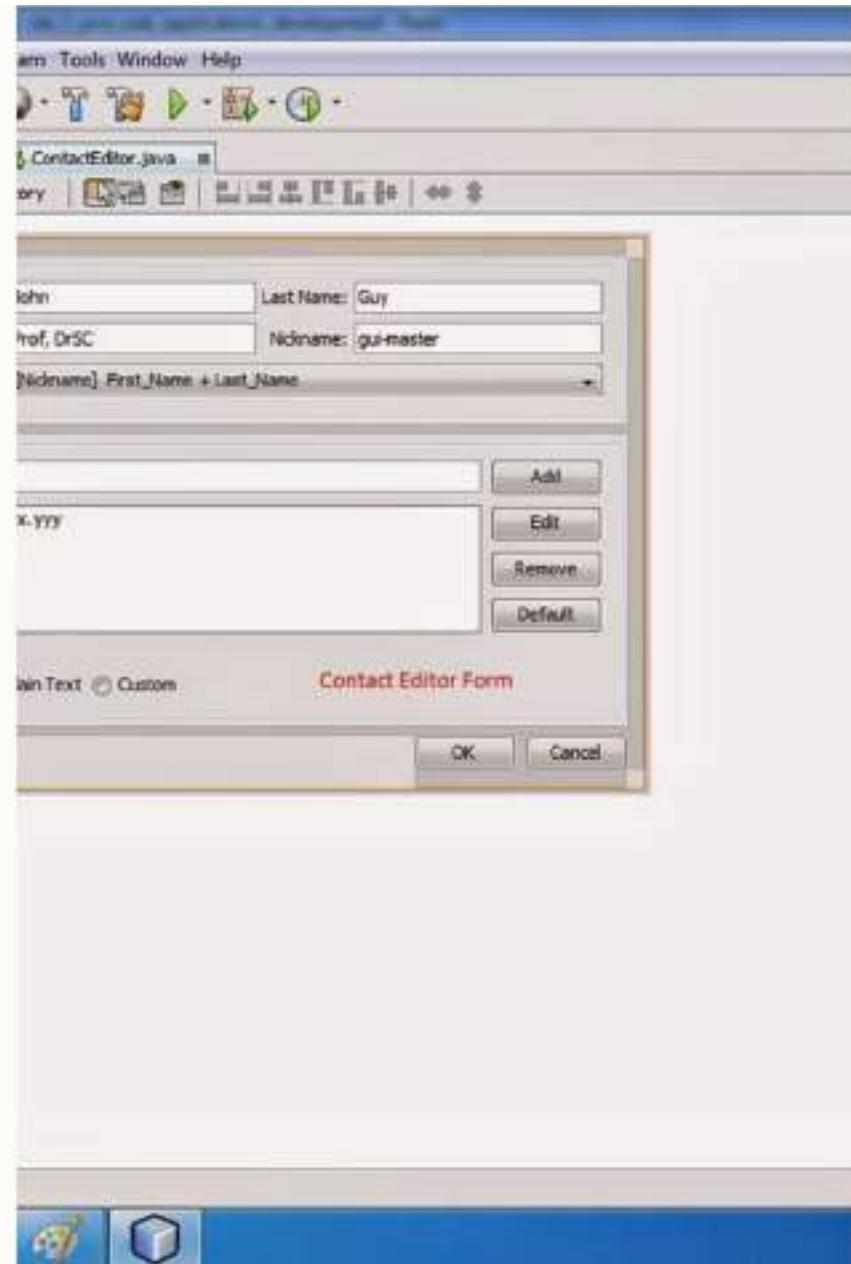
## Separate Panels

Distinct views for admin and user modules improve usability.



## Event Handling

Buttons and forms wired with event listeners for actions.





## Conclusion & Future Enhancements



### Project Outcome

Core functionality demonstrates Java, GUI, and database skills.



### Next Steps

Add online payments, user accounts, and better UI.