



INTERNATIONAL UNIVERSITY - VNU HCMC
FINAL PRESENTATION(2025-2026)
WEB APPLICATION DEVELOPER
COURSE ID: ITCS25IU11

TAB: RENT CAR

1	Vũ Huỳnh Quốc Anh	ITITWE22003	Team member
2	Phạm Thành Trung	ITITIU22170	Team member
3	Vũ Quốc Bảo	ITITWE22107	Team member

ASSOC.PROF : Nguyễn Văn Sinh
LAB ADVISER: Nguyễn Trung Nghĩa

Test date: 24th December 2025

TABLE OF CONTENTS

•Chapter 1. Project Overview

- 1.1. Introduction
- 1.2. Objectives
- 1.3. Target Users

•Chapter 2 Technology Stack

- 2.1. Frontend Technologies
- 2.2. Backend Technologies
- 2.3. Database System
- 2.4. Tools & IDEs

•Chapter 3 System Design

- 3.1. Use Case Diagram
- 3.2. Entity Relationship Diagram (ERD)
- 3.3. Database
- 3.4. System Architecture (UML)

•Chapter 4 Features and Functionalities

- 4.1. User Authentication
- 4.2. Property Listing
- 4.3. Booking Inquiry
- 4.4. Add Property



TABLE OF CONTENTS

•Chapter 5. Implementation

5.1. FrontendImplementation

5.2. BackendImplementation

5.3. JSPPageIntegration.

5.4. MySQLDatabaseConfiguration

•Chapter 6. Testing and Evaluation

6.1. TestCases

6.2. Results



CHAPTER 1: PROJECT OVERVIEW

Find Your Dream Car Today

Explore our wide range of vehicles for rent.

Book now and enjoy the ride!

Browse Cars



1.1 INTRODUCTION

- The "TAB Car Rental" website is a modern online platform developed for TAB – a professional car rental company. This web application allows customers to easily browse, select, and book vehicles directly from our premium fleet. With features such as vehicle search, pick-up and drop-off date selection, and quick pickup, TAB ensures a seamless, convenient, and luxurious car rental experience.

1.2. OBJECTIVES THE PRIMARY GOALS OF THIS PROJECT ARE:

- Simplify the car rental process for customers through an intuitive, centralized website.
- Provide information on a wide range of available models and instant booking confirmation.
- Allow for quick vehicle pickup and electronic contracts.
- Ensure secure transactions and efficient rental management with a robust support system.

1.3. TARGET USERS THE TARGET AUDIENCE INCLUDES:

- Individual customers (travelers, business professionals) seeking reliable car rentals based on location, dates, vehicle type, and budget.
- Corporate clients needing short-term or long-term vehicle rentals for employees or events.
- Tourists and visitors looking for premium, well-maintained cars with excellent customer support.

CHAPTER 2. Technology Stack



2.1

FRONTEND TECHNOLOGIES

- HTML5, Thymeleaf & Bootstrap 5 Purpose: Build a fully responsive and modern user interface with server-side dynamic rendering. Key features: Navigation bar, search/filter for cars, grid layout for car listings, detailed booking forms, login/register pages, date inputs for pick-up/drop-off, and custom green theme (#28a745).
- All interactivity (form submission, validation, page navigation) is handled server-side via Thymeleaf and Spring controllers – no client-side JavaScript is used.
- Thymeleaf integrates seamlessly with Spring Boot for conditional rendering, loops (e.g., car lists), and form binding.

2.2

BACKEND TECHNOLOGIES

- Spring Boot (Core framework) Purpose: Provides auto-configuration, embedded Tomcat server, and clean layered architecture for efficient development.
- Key components:
 - Controllers (BookingController, CarController, AuthController, HomeController, UserController) – Handle requests and return Thymeleaf views.
 - Services (UserServiceImpl, CarServiceImpl, BookingServiceImpl) – Contain business logic (e.g., check car availability, calculate total price).
 - Repositories (Spring Data JPA) – Data access layer.
 - Entities (User, Car, Booking) – Domain models with relationships.
- Spring Security: Handles authentication, authorization, and role-based access (USER/ADMIN).
- DTOs: For safe data transfer between layers (e.g., BookingRequestDTO).

2.3

DATABASE

- MySQL (via Spring Data JPA & Hibernate) Purpose: Persistent storage for users (with roles), cars (name, type, pricePerDay, image, availability), and bookings (dates, status, totalPrice).
- Initial data loaded via data.sql (5–6 sample cars + test accounts).

CHAPTER 3 SYSTEM DESIGN

3.1. USE CASE DIAGRAM

Actors

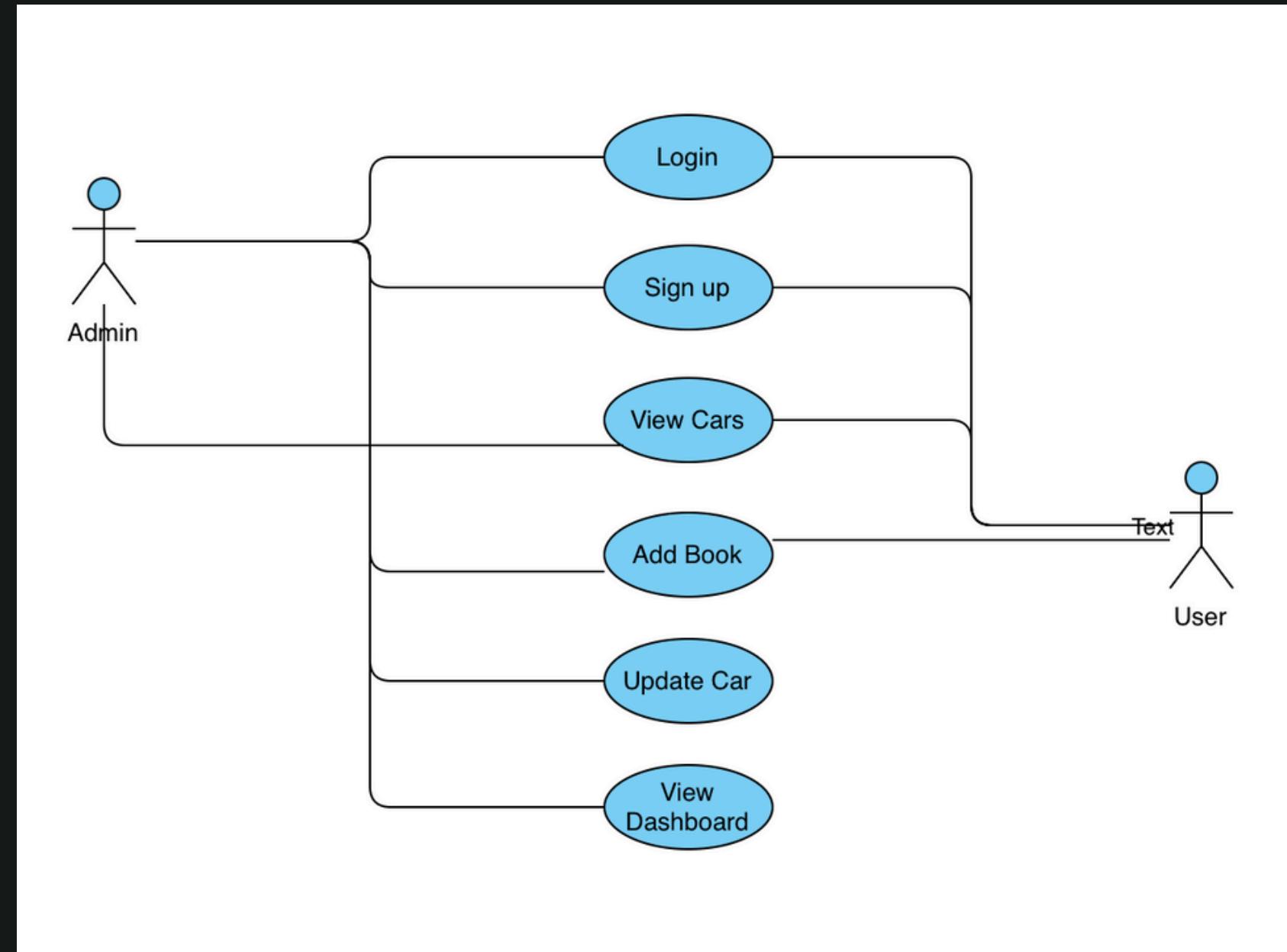
- User: Customers (guests or logged-in) – browse cars, register, login, make bookings, view personal bookings.
- Admin: Administrator – manage cars (add/update/remove), manage all bookings.

Main Use Cases

- Sign Up / Register: Create new account (name, email, phone, password).
- Login: Authenticate to access booking features.
- View Cars: Browse available cars (available to guests).
- View Car Details (extends View Cars): See specific car info (image, type, price).
- Make Booking / Add Book: Select dates and submit booking (requires login).
- View Dashboard: Check personal booking history.
- Update Car: Admin edits car details.
- Manage Bookings: Admin views/updates all bookings.

[Home](#)[Service](#)[About Us](#)[Contact](#)

—



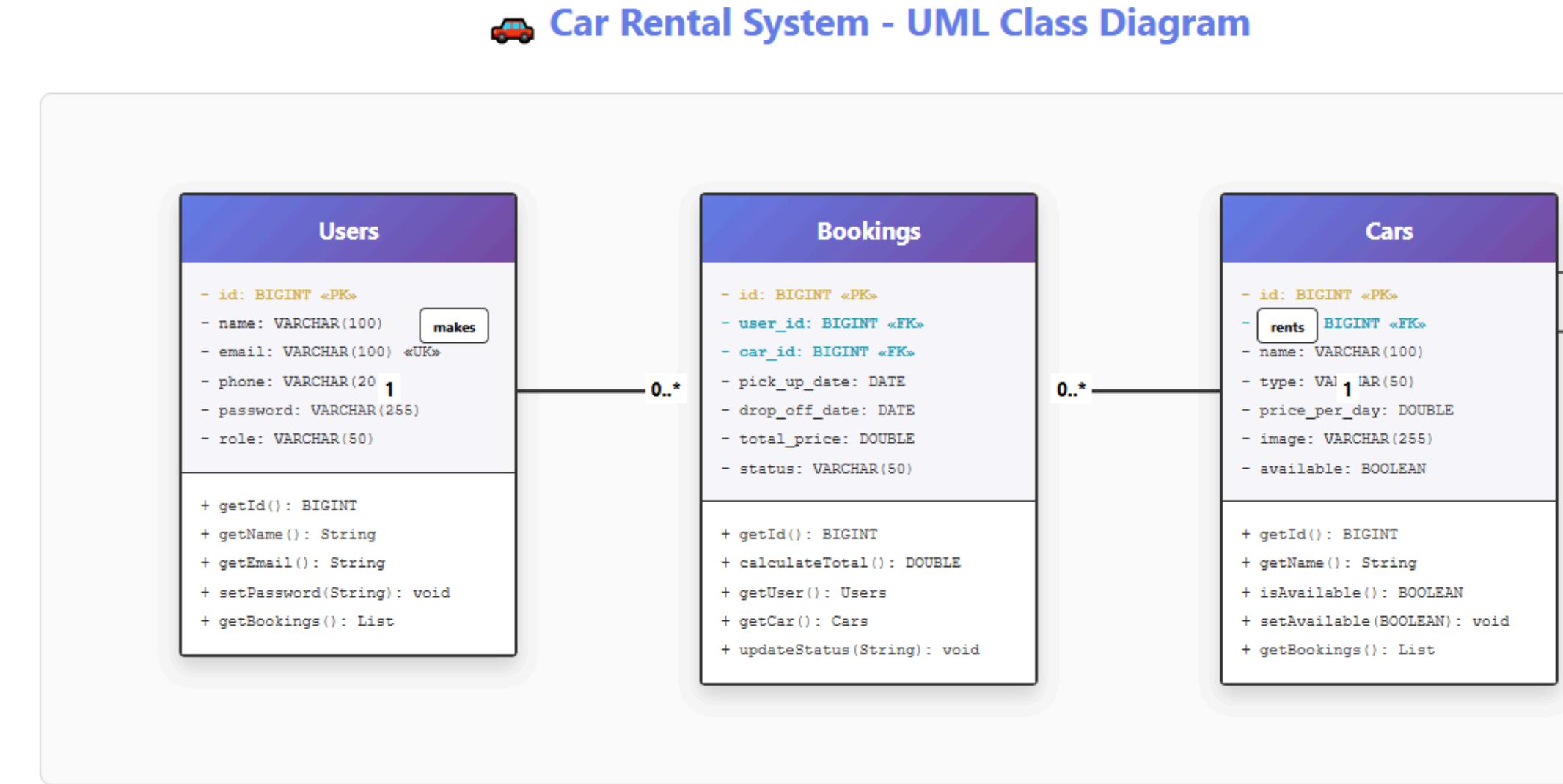
CHAPTER 3 SYSTEM DESIGN

3.3. Database

```
• INSERT INTO users (name, email, phone, password, role)
VALUES
  ('Admin', 'admin@gmail.com', '0900000000',
   '$2a$10$7QF1E6IYw0gU6Zp5yYFJKeNWh8s4hD5G1nJx4nY5Ck8Q7zP8QFZ1a',
   'ROLE_ADMIN'),
  ('Test User', 'user@gmail.com', '0911111111',
   '$2a$10$7QF1E6IYw0gU6Zp5yYFJKeNWh8s4hD5G1nJx4nY5Ck8Q7zP8QFZ1a',
   'ROLE_USER');

• INSERT INTO cars (name, type, price_per_day, image, available)
VALUES
  ('BMW S1000RR', 'Moto', 180, '/images/cars/s1000rr.jpg', TRUE),
  ('Kia Carnival', 'MPV', 120, '/images/cars/carnival.jpg', TRUE),
  ('Toyota Hilux', 'Pickup', 95, '/images/cars/hilux.jpg', TRUE),
  ('Audi R7', 'Sports', 220, '/images/cars/r7.jpg', TRUE),
  ('Ford Transit', 'Van', 150, '/images/cars/transit.jpg', TRUE),
  ('Toyota Vios', 'Sedan', 65, '/images/cars/vios.jpg', TRUE),
  ('Xpander', 'MPV', 90, '/images/cars/xpander.jpg', TRUE);
```

3.4. SYSTEM ARCHITECTURE (UML)

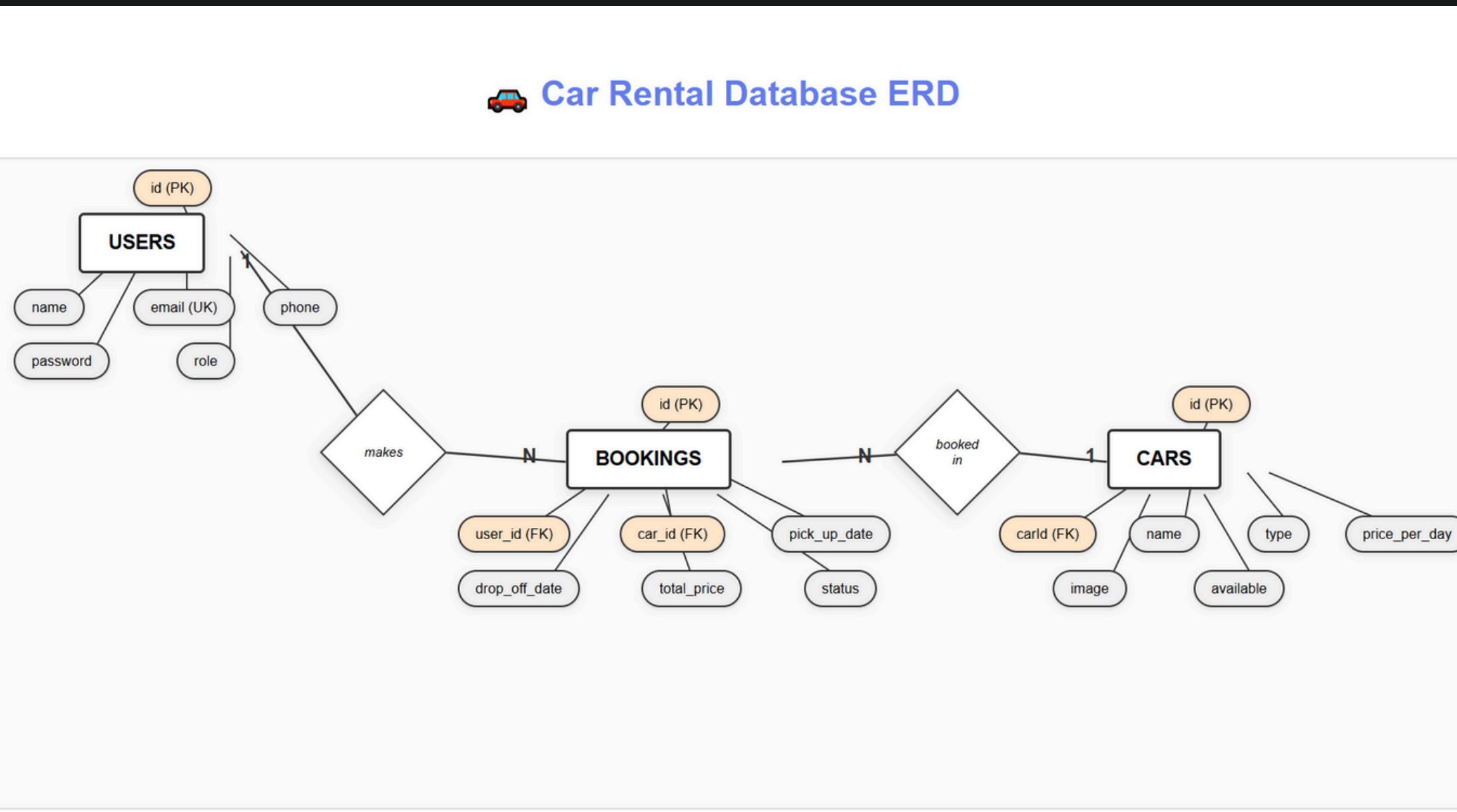


CHAPTER 3 SYSTEM DESIGN

[Home](#)[Service](#)[About Us](#)[Contact](#)

—

3 . 2 . ENTITY RELATIONSHIP DIAGRAM ERD



Main Entities and Attributes

- **USERS**
 - Attributes: id (PK), name, email (UNIQUE), phone, password, role (USER or ADMIN)
- **CARS**
 - Attributes: id (PK), name, type, price_per_day, image, available (BOOLEAN, default TRUE)
- **BOOKINGS**
 - Attributes: id (PK), user_id (FK), car_id (FK), pick_up_date, drop_off_date, total_price, status

Relationships

- One USER makes many BOOKINGS (1-to-many).
- One CAR is booked in many BOOKINGS (1-to-many).
- Each BOOKING belongs to exactly one USER and one CAR (foreign keys: user_id → USERS.id, car_id → CARS.id).

This simple and efficient schema supports core operations: user authentication with roles, car inventory management, and secure booking tracking while maintaining data integrity through primary and foreign key constraints.



LUXURY



CHAPTER 4 FEATURES AND FUNCTIONALITIES

4.1. User Authentication

- Users are associated with a `registerUser` object; users must log in to book a car. A `registerUser` can book one car. Users can view cars without logging in.

4.1. USER AUTHENTICATION -LOGIN PAGE

The image displays two side-by-side versions of a user authentication login page. Both versions feature a central input field for 'Email' and a separate input field for 'Password'. A large green 'Login' button is positioned below the password field. At the bottom of each page, there is a link 'No account? Register'.

Left Side (Dark Mode):

- Background is dark.
- Text color: white.
- Input fields have a dark background.
- Buttons are dark green.

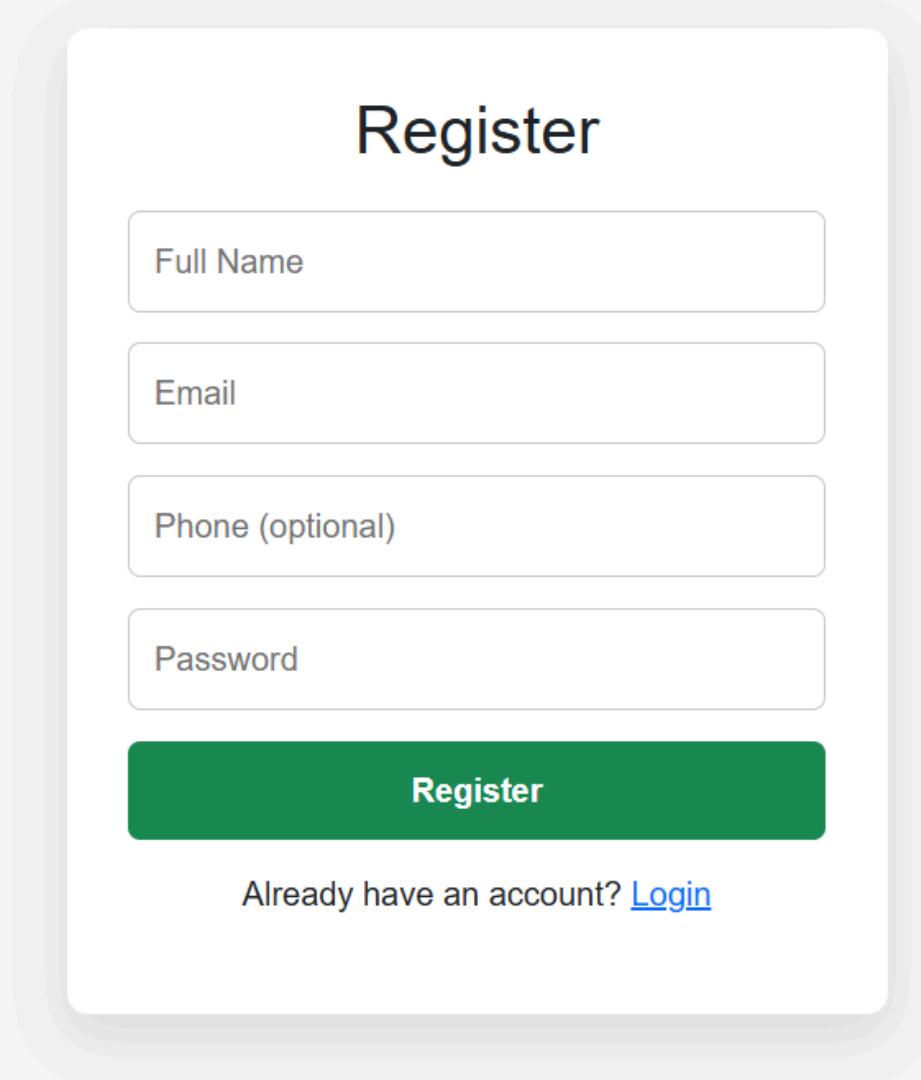
Right Side (Light Mode):

- Background is light gray.
- Text color: black.
- Input fields have a light gray background.
- Buttons are green.

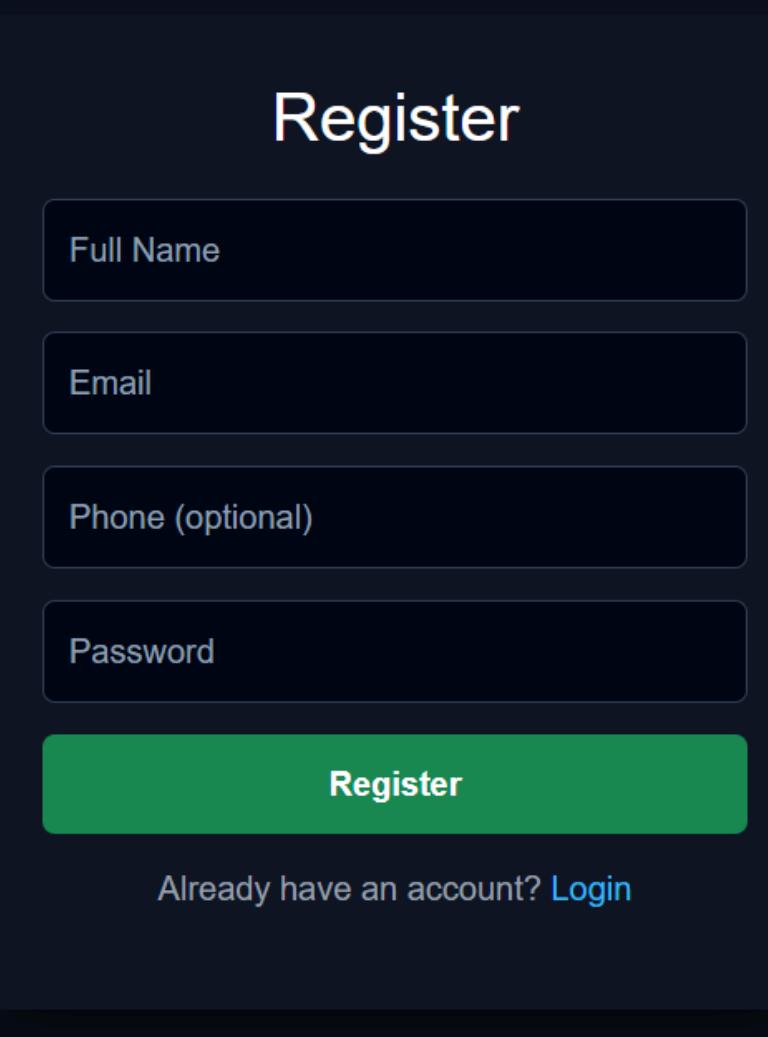
Common Elements:

- Login Title:** 'Login' centered at the top of both forms.
- Email Input:** Labeled 'Email' with a corresponding input field.
- Password Input:** Labeled 'Password' with a corresponding input field.
- Login Button:** A large green button labeled 'Login'.
- Registration Link:** 'No account? [Register](#)' located at the bottom of both forms.

4.1. USER AUTHENTICATION -REGISTER PAGE



The image shows a light-colored mobile application interface for a registration form. At the top, the word "Register" is displayed in a bold, black font. Below it are four input fields: "Full Name", "Email", "Phone (optional)", and "Password", each enclosed in a white rectangular box with a thin gray border. At the bottom of the form is a large green rectangular button with the word "Register" in white. Below the button, the text "Already have an account? [Login](#)" is displayed in a smaller, dark font.



The image shows the same registration form as the first one, but in dark mode. The background is black, and the entire form is contained within a dark gray rounded rectangle. The title "Register" is at the top in white. Below it are the four input fields: "Full Name", "Email", "Phone (optional)", and "Password", each in a white rectangular box with a thin gray border. At the bottom is a large green rectangular button with the word "Register" in white. Below the button, the text "Already have an account? [Login](#)" is displayed in a smaller, light blue font.

4.2. PROPERTY LISTING

Our Car Collection



BMW M3

\$180 /day

Book Now



Kia Carnival

\$120 /day

Book Now



Toyota Hilux

\$95 /day

Book Now



Audi R7

\$220 /day

Book Now



Ford Transit 16 chỗ

\$150 /day

Book Now



Toyota Vios

\$65 /day

Book Now

4.2. PROPERTY LISTING

Our Car Collection



BMW S1000RR
\$180.0 /day

[Book Now](#)



Kia Carnival
\$120.0 /day

[Book Now](#)



Toyota Hilux
\$95.0 /day

[Book Now](#)



Audi R7
\$220.0 /day

[Book Now](#)



Ford Transit
\$150.0 /day

[Book Now](#)



Toyota Vios
\$65.0 /day

[Book Now](#)

[← Back to Cars](#)

Book Your Car

Toyota Hilux

Pickup

95.0 \$/day

Personal Information

First Name *

Thành Trung

Last Name *

Phạm

Contact Information

Email Address

ttrung1326@gmail.com

Phone Number *

+84394674592

Rental Details

Pick-up Location *

Ba

Drop-off Location *

Ria

Pick-up Date *

12/25/2025

Drop-off Date *

12/26/2025

Additional Notes (Optional)

Any special requests or requirements?

4.4 Confirm Information

The screenshot shows a web browser window with the URL <localhost:8086/booking/payment>. The browser tabs include Facebook, Black Elegant Interior Design Prese, Payment Method, and the current tab. The main content area is titled "Payment Method".

Booking Summary:

Car	Price/Day	Pick-up Date	Drop-off Date
Toyota Hilux	\$95.0	2025-12-25	2025-12-26

Customer: Thành Trung Phạm
Phone: +84394674592
Email: ttrung1326@gmail.com

Total Days: 1
Total Price: \$95.0

Additional Notes:

Cash Payment:

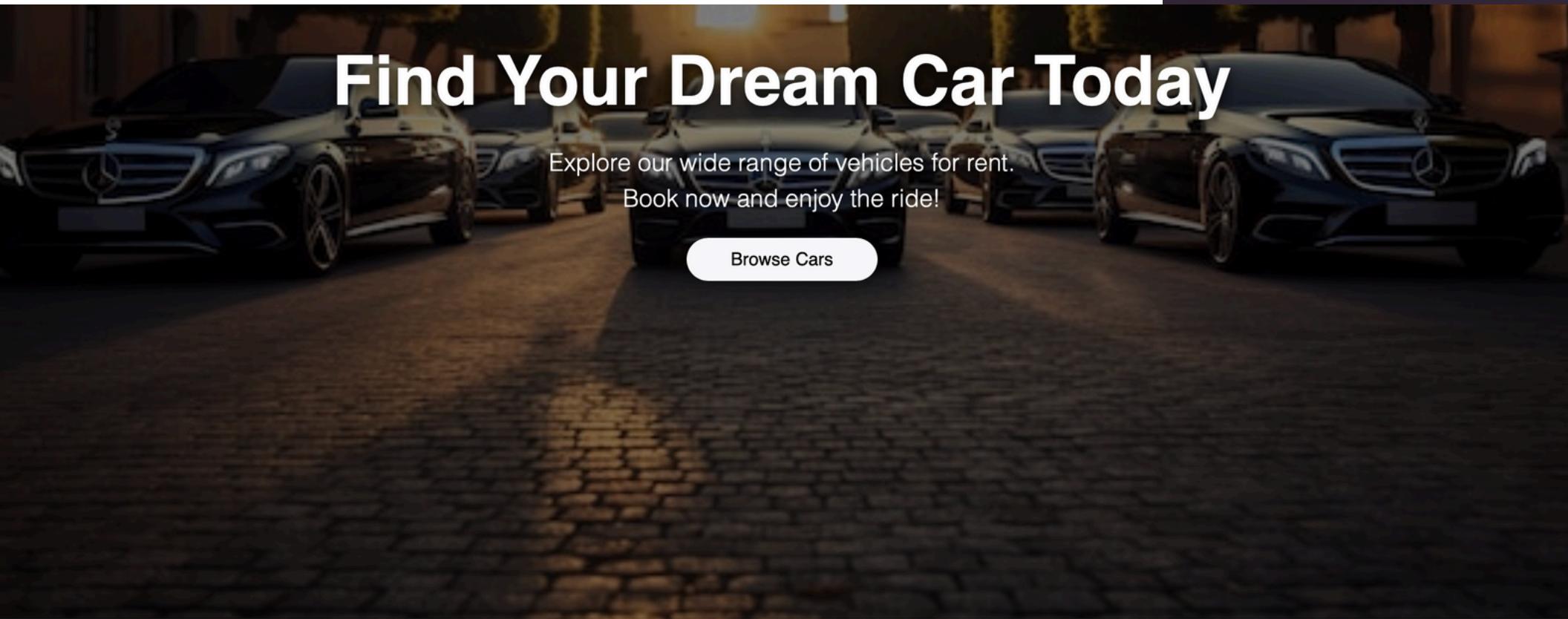
You will pay in cash when you pick up the vehicle at our location. Please bring the exact amount or we can provide change.

Important Information:

- Payment must be made in cash upon vehicle pick-up
- Please bring a valid driver's license
- A security deposit may be required

Confirm Booking

CHAPTER 5 USER AND ACCOUNT SYSTEM



5.1. FRONTEND IMPLEMENTATION

The frontend of the TAB Self-Driving Car Rental Website was designed to provide an intuitive and fully responsive user interface for customers and administrators. All pages are rendered server-side using Thymeleaf templates integrated with Spring Boot, ensuring fast loading and SEO-friendly structure without relying on client-side JavaScript.

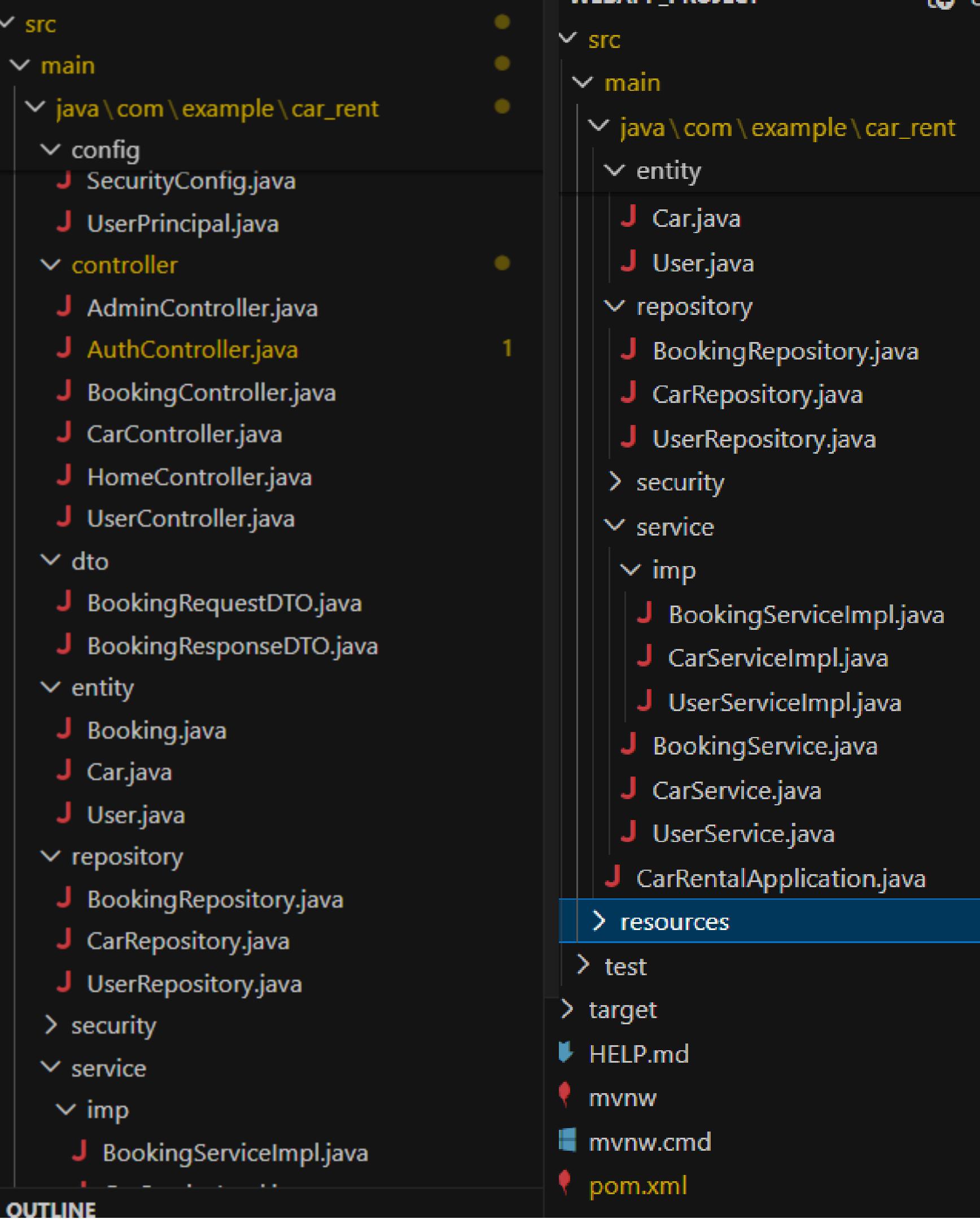
TECHNOLOGIES USED

- **HTML (HyperText Markup Language)**
- **CSS (Cascading Style Sheets)**
- **Springboot**

Structure and Design

- **Consistent layout using fragments: header.html, navbar.html, footer.html, scripts.html**
- **Navigation bar with links to Home, Cars, Login/Register, Dashboard (for logged-in users)**
- **Mobile-friendly design with Bootstrap grid system**
- **Form validation handled server-side**

5.2. BACKEND IMPLEMENTATION



5.3. JSP PAGE INTEGRATION

```
✓ resources
  ✓ static
    ✓ css
      # booking.css
      # dashboard.css
      # home.css
      # navbar.css
      # payment.css
      # style.css
      # success.css
      # theme.css
    > images
    > js
    ✓ templates
      ✓ admin
        ⚡ dashboard.html
      ✓ auth
        ⚡ login.html
        ⚡ register.html
      ✓ booking
        ⚡ booking-form.html
        ⚡ payment.html
        ⚡ success.html
      ✓ fragments
        ⚡ navbar.html
        ⚡ cars.html
        ⚡ home.html
      ⚡ application.properties
```

5.4. MYSQL DATABASE CONFIGURATION

Configuring a MySQL database for an online car rental application supports storing and managing user data, vehicles, fares, etc., ensuring robust data processing for the application's core functionalities. Database setup.

Database Name: rentcar_db.
Connection: Hosted locally (localhost) with secure credentials (e.g., username: root, password: set by admin).

```
8  -- =====
9  -- USERS TABLE
10 -- =====
11 • ⊖ CREATE TABLE users (
12     id BIGINT AUTO_INCREMENT PRIMARY KEY,
13     name VARCHAR(100) NOT NULL,
14     email VARCHAR(100) NOT NULL UNIQUE,
15     phone VARCHAR(20),
16     password VARCHAR(255) NOT NULL,
17     role VARCHAR(50) NOT NULL
18 );
19 -- =====
20 -- CARS TABLE
21 -- =====
22 • ⊖ CREATE TABLE cars (
23     id BIGINT AUTO_INCREMENT PRIMARY KEY,
24     carId BIGINT NULL,
25     name VARCHAR(100) NOT NULL,
26     type VARCHAR(50),
27     price_per_day DOUBLE NOT NULL,
```

```
33
34 -- =====
35 -- BOOKINGS TABLE
36 -- =====
37 • ⊖ CREATE TABLE bookings (
38     id BIGINT AUTO_INCREMENT PRIMARY KEY,
39     user_id BIGINT NOT NULL,
40     car_id BIGINT NOT NULL,
41     pick_up_date DATE NOT NULL,
42     drop_off_date DATE NOT NULL,
43     total_price DOUBLE NOT NULL,
44     status VARCHAR(50) NOT NULL,
45     CONSTRAINT fk_booking_user FOREIGN KEY (user_id) REFERENCES users(id),
46     CONSTRAINT fk_booking_car FOREIGN KEY (car_id) REFERENCES cars(id)
47 );
48
```

CHAPTER 6. CHALLENGES AND SOLUTIONS



6.1. Session Management

Managing user sessions to maintain authentication across multiple pages proved challenging, especially with different user roles (USER and ADMIN). Early attempts led to frequent unintended logouts due to improper handling of session state in a server-side rendered application.

Solution A robust session management system was implemented using Spring Security integrated with Spring Boot.

- Authentication is handled via form-based login (`login.html`) with username/password validation.
- Upon successful login, Spring Security creates an authenticated `SecurityContext` stored in the `HttpSession`.
- Role-based access control (`@PreAuthorize` or secured paths) ensures USER can only access booking features and dashboard, while ADMIN can manage cars and all bookings.
- Session persistence is maintained server-side across page navigations (Thymeleaf templates), eliminating the need for client-side tokens or JavaScript. This approach ensures secure, reliable authentication and significantly improves user experience with seamless navigation after login.

THANK YOU FOR LISTENING

Find Your Dream Car Today

Explore our wide range of vehicles for rent.

Book now and enjoy the ride!

[Browse Cars](#)