PROJECT

Topic: Car Rentals

Design and Implementation of a Car Rental System

PySharks

July 7, 2025

Contents

1 Introduction

This section introduces the car rental system, its purpose, and its target users. It briefly describes the problem it solves and its significance in the market.The Car Rental System streamlines car rental company operations, managing vehicle fleet, purchasing to fleet and bookings. Customers can easily browse, reserve, and rent cars for specific durations.The system facilitates seamless rental reservations, enabling customers to get the vehicles based on preferences.

Figure 1: Rental Cars

2 Business Domain Description

The Car Rental System is a software application designed to manage the operations of a car rental company.It handles various aspects of the business, including vehicle management, customer management and booking.

2.1 Description:

The system allows customers to browse through the available vehicles, make reservations, and rent cars for a specified duration. It also enables the car rental company to manage its fleet of vehicles, track rentals, generate invoices, and maintain customer records.

2.2 Key Features:

2.2.1 Vehicle Management

Add vehicles to the fleet

2.2.2 Customer Management

Create and manage customer accounts Store customer personal information and rental history

2.2.3 Rental Reservation:

Allow customers to get the available vehicles based on criteria Provide options for customers to reserve and rent vehicles

2.2.4 Booking and Payment

Calculate rental fees based on rental duration, vehicle type, and additional services.Process payments like googlepay,Paypal and bank transfer

3 Use cases

3.1 Adding a Vehicle

The car rental company purchases a new vehicle and needs to add it to the fleet.

3.1.1 Steps

Enter the details of the vehicle such as make, model, year, registration number, etc. The system updates the fleet database with the new vehicle information.

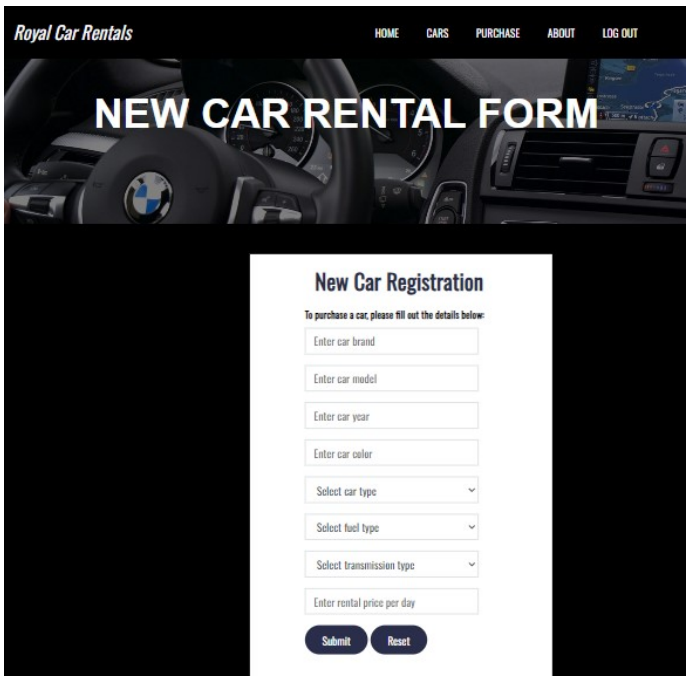

Figure 2: Add new Rental Cars

3.2 Customer Reservation

Description: A customer wants to reserve a car for a weekend trip.

3.2.1 Steps

The User browse available vehicles based on location, date range, and vehicle type. After finding a suitable option, they select it and proceed to make a reservation. They provide the required details such as pickup location, drop-off location, and rental duration.

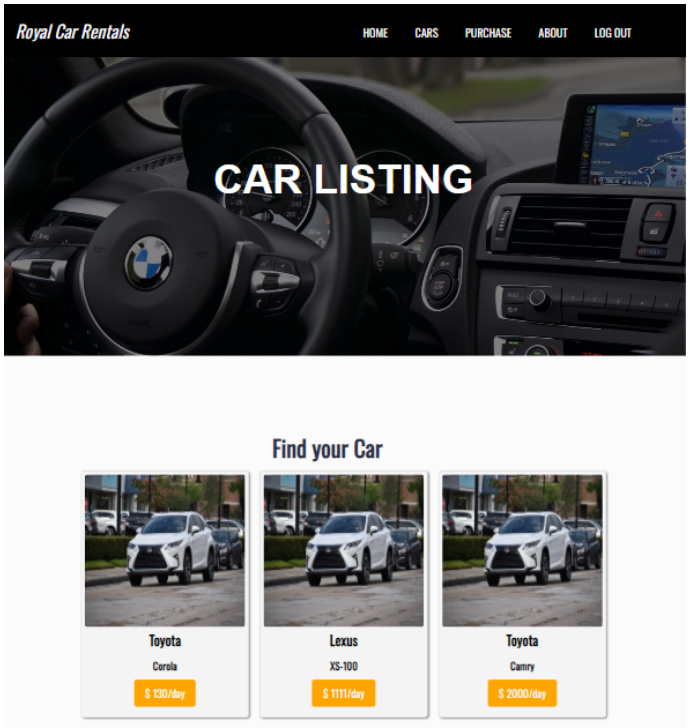

Figure 3: Listing all the Cars

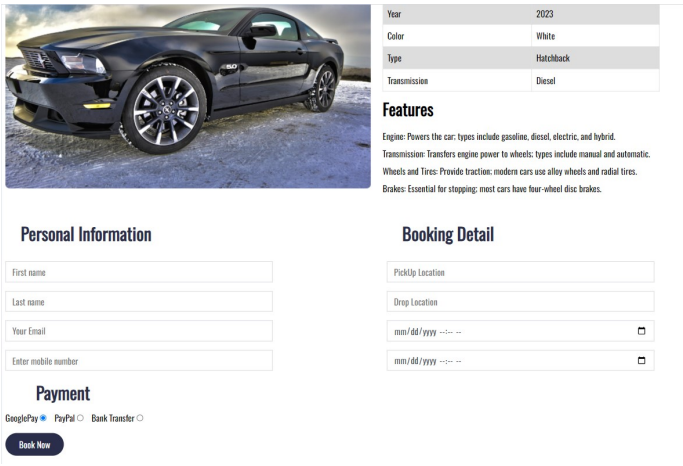

Figure 4: Booking the Cars

3.3 User Creation

Creation of new user account.

3.3.1 Steps

Creating new user account using username and password

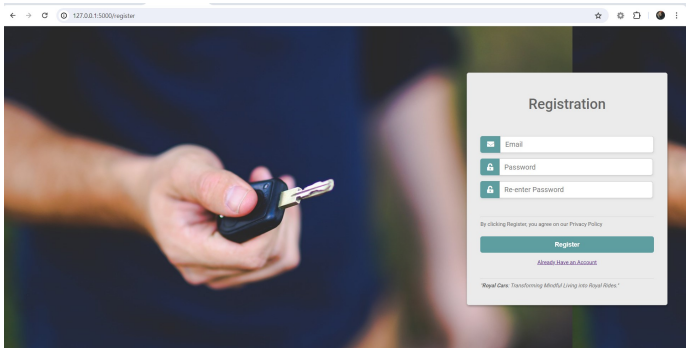

Figure 5: Creating User Account

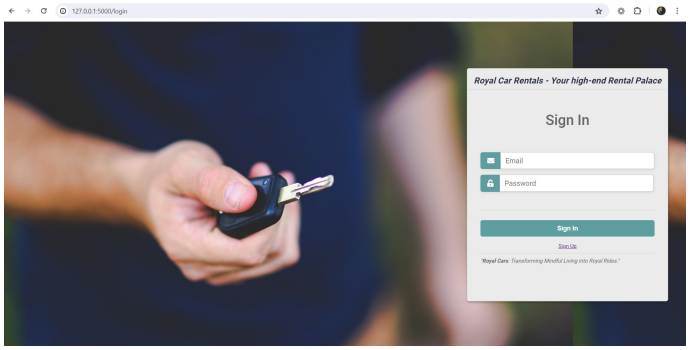

Figure 6: Login using user details

4 Business Objects

4.1 cardata

Attributes:

- Brand
- Model
- Year
- Color
- Type
- Fuel
- Transmission
- RentalPrice

Operations:

- AddNewVehicles
- DisplayAllVehicles

4.2 Bookings

Attributes:

- referenceNumber
- FirstName
- Lastname
- Email
- Mobile
- PickupLocation
- DropOffLocation
- PickupDate
- DropoffDate
- Payment
- Sold

Operations:

- Addnewbookings

4.3 userRegistration

Attributes:

- email
- password
- active

5 Important Links

5.1 Overleaf Editable Link

5.1.1 https://www.overleaf.com/6627733671nwpkwzfjqvpy5a8d8a

5.2 Google Sites URL

5.2.1 https://sites.google.com/view/royalcarrentalpysharks/home

5.3 YouTube Video LINK

5.3.1 https://youtu.be/$_KMnqNlgjWk$

5.4 Docker Link

5.4.1 docker pull 8075997171/royalcarrentals:V1.1

5.5 Git Link

5.5.1 https://github.com/Jincy-1234/CarRentals.git

6 System Overview

6.1 Design

6.1.1 Frontend Components:

- The system have multiple HTML templates (index.html, about.html, register.html, login.html, cars.html, home.html, booking.html) rendered using pyflask.
- These templates likely provide the user interface for different pages like home, about, registration, login, car listing, booking, etc.

6.1.2 Backend Components:

- Flask: The Flask web framework is used to handle HTTP requests and responses. MongoDB Database: MongoDB is used as the database to store data related to car rental information, user registration, and bookings. The pymongo library is used to interact with the MongoDB database.
- User Registration: Users can register via the /register route. The data is stored in the userRegistration collection in the MongoDB database.
- User Authentication: There's a basic check for existing users via the /checkuserexistence route, which returns a response to determine user existence.
- Car Data Management: Car rental data such as brand, model, year, color, etc., is stored in the carRentalData collection in the MongoDB database. Routes like /data and /carsdata handle the insertion and retrieval of car data.
- Booking Management: Users can book cars via the /book route. Booking data, including a randomly generated reference number, is stored in the bookings collection in the MongoDB database.

- Random Number Generation: A function generatereferencenumber generates a random reference number for booking identification.

6.1.3 Overall Architecture:

- The application follows a client-server architecture where the Flask app serves as the backend server handling HTTP requests from clients (browsers).
- MongoDB serves as the database backend, storing user data, car rental data, and booking information.
- The frontend (HTML templates) interacts with the backend through HTTP requests, enabling users to register, log in, view available cars, and make bookings.

6.2 User Interface

6.2.1 Homepage (index.html):

- The homepage likely includes a navigation bar with links to different sections of the website, such as "Home", "About", "Register", "Login", "Cars", and "Booking".
- It may also feature promotional banners, images, or car rental deals to attract users' attention.
- Users may find a search bar or filters to quickly search for cars based on their preferences.

6.2.2 About Page (about.html):

- This page may provide information about the car rental company, its history, mission, vision, and any unique selling propositions.
- It could include images or graphics related to the company or its services. Registration Page (register.html):
- Users are presented with a form to register for an account. Fields likely include email, password, and possibly additional information like name, contact details, etc.
- There might be validation checks to ensure the entered information is correct.

6.2.3 Login Page (login.html):

- Users can log in with their registered email and password.
- There may be options for password recovery or account activation if the account is not yet activated.

6.2.4 Car Listing Page (cars.html):

- Displays a list of available cars for rental.
- Each car may have its image, brand, model, year, color, type, fuel type, transmission, and price displayed.
- Users may have the option to filter or sort cars based on criteria such as brand, model, price, etc.

6.2.5 Booking Page (booking.html):

- Users can select a car they want to rent and proceed with the booking process. The page may contain a form where users enter their details such as name, contact information, pickup/drop-off locations, dates, etc.
- Users may also see a summary of their booking before confirming.

6.3 User Experience (UX):

6.3.1 Registration and Login:

- Users can easily register for an account by providing necessary details and can log in subsequently.
- Clear instructions and error messages guide users through the registration and login process.

6.3.2 Finding Cars:

- The car listing page presents available cars in a clear and organized manner, making it easy for users to browse and compare.
- Filters and sorting options help users narrow down their search based on their preferences.

6.3.3 Making a Booking:

- The booking process should be intuitive and straightforward.
- Users should be guided step-by-step through the booking form, with clear labels and instructions for each field.
- Error handling should be in place to notify users of any mistakes or missing information.

7 Conclusion

The car rental system project is a Flask-based web application designed to facilitate the rental process for users. It includes features such as user registration, car listing, booking, and user authentication. MongoDB is used as the backend database to store user data, car rental information, and booking details.