Online Car Rental Platform – Project Write-Up

1. Introduction

This project implements an online car rental platform using Object-Oriented Programming (OOP) in Python. The system allows customers to view available cars, rent cars on an hourly, daily, or weekly basis, and return with an automatically generated bill. The solution is split into two files: a Python module (car_rental.py) containing the core classes and logic, and a Jupyter Notebook (main.ipynb) that provides the user interface and runs the main program.

---

2. Objectives

- To practice OOP concepts by modelling real-world entities (rental company, customer).
- To create a modular, maintainable Python application.
- To handle inventory management, rental requests, returns, and billing.
- To ensure robust input validation and clear user prompts.

3. Design & Implementation

A. car_rental.py

This module contains two main classes:

CarRental

- **Purpose:** Manages the car inventory and handles all rental and return operations.
- **Key Methods:**
- display_available_cars(): Shows the number of cars currently available.
- rent_hourly(num_cars), rent_daily(num_cars), rent_weekly(num_cars): Rent cars for different periods, validating input and updating inventory.
- _rent_car(num_cars, rental_type): Shared logic for all rental types; stores rental time and type.
- return_car(rental_info): Processes car returns, updates inventory, and calculates the bill.

- calculate_bill(rental_time, rental_type, num_cars): Computes the bill based on rental duration and rate.

Customer

- **Purpose:** Represents a customer, tracks their current rental, and interacts with the rental company.
- **Key Methods:**
- request_car(car_rental, rental_type, num_cars): Requests a rental from the company, ensuring only one active rental at a time.
- return_car(car_rental): Returns the rented car(s) and receives the bill.

B. main.ipynb

- **Imports** the car_rental module and initializes the system.
- **Prompts** the user for their name and creates a Customer object.
- **Displays** a menu allowing the user to:

1. View available cars
2. Rent cars (hourly/daily/weekly)
3. Return cars
4. Exit

- **Handles** each option by calling the appropriate class methods and displaying informative messages.
- **Validates** all user inputs and provides error feedback when necessary.
- **Loops** until the user chooses to exit.

4. Key Features

- **Object-Oriented:** Uses classes for clear separation of responsibilities.
- **Inventory Management:** Tracks and updates available cars.
- **Flexible Rentals:** Supports hourly, daily, and weekly rentals.
- **Automated Billing:** Calculates bills based on actual usage.
- **Input Validation:** Handles invalid entries gracefully.
- **User-Friendly Interface:** Clear instructions and feedback throughout.

5. Testing

The system was tested for:

- Renting and returning cars under all rental modes.
- Attempting to rent more cars than available.
- Handling invalid inputs (e.g., negative numbers, non-numeric input).
- Preventing multiple simultaneous rentals by the same customer.
- Accurate billing for various durations.

6. Possible Enhancements

- Support for multiple customers.
- Persistent storage of rental records.
- Different car models and rates.
- Discount logic for long-term rentals.
- Graphical or web-based user interface.

7. Conclusion

This project successfully meets the requirements for an online car rental platform using OOP in Python. It demonstrates effective use of classes, robust input handling, and clear user interaction. The modular structure makes it easy to extend and maintain.