Project Title: Online Car Rental Platform

Project Objective:

Build an online car rental platform using Object-Oriented Programming in Python.

Problem Statement:

A car rental company has requested you to build an online car rental platform where customers should be able to view the available cars that can be rented on an hourly, daily, or weekly basis. The company can display the available inventory and confirm requests by checking the available stock. Customers will receive an auto-generated bill when they return the car.

For simplicity, let's assume that:

1. Customers can rent cars from any one of the following options—hourly, daily, or weekly rental.
2. Customers are free to choose any number of cars they want, provided the number of available cars is more than the number of requested cars.

You must use the following tools:

Jupyter Notebook: To create the module and main project files

Instructions to Perform:

1. Create a module (.py file) for car rental and import the built-in module DateTime to handle the rental time and bill.
2. Create a class for renting the cars and define a constructor in it.
3. Define a method for displaying the available cars. Also, define methods for renting cars on an hourly, daily and weekly basis, respectively.
4. Inside these methods, make sure that the number of requested cars is positive and lesser than the total available cars.
5. Store the time of renting a car in a variable, which can later be used in the bill while returning the car.
6. Define a method to return the cars using rental time, rental mode (hourly, daily, or weekly), and the number of cars rented.
7. Inside the return method; update the inventory stock, calculate the rental period, and generate the final bill.

8. Create a class for customers and define a constructor in it.
9. Define methods for requesting the cars and returning them.
10. Next, create the main project (.ipynb) file and import the car rental module in it.
11. Define the main method and create objects for both car rental and customer classes.
12. Inside the main method, take the customer’s input as a choice for displaying car availability, rental modes, or returning the cars.
13. Use the relevant method for the customer’s input and print relevant messages.
14. Run the main method to start your project.

Project Code with Output

Code:

1.(Car_rental.py):

```
import datetime

class CarRental:
    def __init__(self, stock=0):
        self.stock = stock

    def display_cars(self):
        print(f"Available cars: {self.stock}")
        return self.stock

    def rent_hourly(self, num_of_cars):
        if num_of_cars <= 0:
            print("Number of cars should be positive!")
            return None
        elif num_of_cars > self.stock:
            print("Sorry! Not enough cars available.")
            return None
        else:
            rental_time = datetime.datetime.now()
            self.stock -= num_of_cars
            print(f"{num_of_cars} car(s) rented on hourly basis at {rental_time.hour} hours.")
            return rental_time

    def rent_daily(self, num_of_cars):
        if num_of_cars <= 0:
            print("Number of cars should be positive!")
            return None
        elif num_of_cars > self.stock:
            print("Sorry! Not enough cars available.")
            return None
        else:
            rental_time = datetime.datetime.now()
            self.stock -= num_of_cars
            print(f"{num_of_cars} car(s) rented on daily basis at {rental_time.day}
day.")
            return rental_time

    def rent_weekly(self, num_of_cars):
        if num_of_cars <= 0:
            print("Number of cars should be positive!")
            return None
        elif num_of_cars > self.stock:
            print("Sorry! Not enough cars available.")
            return None
        else:
            rental_time = datetime.datetime.now()
            self.stock -= num_of_cars
            print(f"{num_of_cars} car(s) rented on weekly basis at
{rental_time.weekday()} day of the week.")
            return rental_time

    def return_cars(self, request):
        rental_time, rental_basis, num_of_cars = request
        bill = 0

        if rental_time and rental_basis and num_of_cars:
            now = datetime.datetime.now()
            rental_period = now - rental_time

            if rental_basis == 1:  # hourly
                bill = round(rental_period.seconds / 3600) * 5 * num_of_cars
            elif rental_basis == 2:  # daily
                bill = round(rental_period.days) * 20 * num_of_cars
            elif rental_basis == 3:  # weekly
                bill = round(rental_period.days / 7) * 60 * num_of_cars

            self.stock += num_of_cars
            print(f"Total bill: ${bill}")
            return bill
        else:
            print("Invalid return request.")
            return None
```

2.(customer.py):

```
class Customer:
    def __init__(self):
        self.cars_rented = 0
        self.rental_basis = 0
        self.rental_time = None

    def request_cars(self):
        cars = input("How many cars would you like to rent?")
        try:
            cars = int(cars)
        except ValueError:
            print("Number of cars should be a positive integer.")
            return -1

        if cars < 1:
            print("Invalid number of cars!")
            return -1
        else:
            self.cars_rented = cars
            return self.cars_rented

    def return_cars(self):
        if self.rental_time and self.rental_basis and self.cars_rented:
            return self.rental_time, self.rental_basis, self.cars_rented
        else:
            return 0, 0, 0
```

3.(main.ipynb):

```
from car_rental import CarRental
from customer import Customer

def main():
    rental_shop = CarRental(10)  # assume 10 cars in stock
    customer = Customer()

    while True:
        print("""
        ====== Car Rental Shop ======
        1. Display available cars
        2. Rent cars on hourly basis ($5 per hour)
        3. Rent cars on daily basis ($20 per day)
        4. Rent cars on weekly basis ($60 per week)
        5. Return rented cars
        6. Exit
        """)

        choice = input("Enter your choice: ")

        if choice == "1":
            rental_shop.display_cars()

        elif choice == "2":
            customer.request_cars()
            rental_time = rental_shop.rent_hourly(customer.cars_rented)
            if rental_time:
                customer.rental_time = rental_time
                customer.rental_basis = 1

        elif choice == "3":
            customer.request_cars()
            rental_time = rental_shop.rent_daily(customer.cars_rented)
            if rental_time:
                customer.rental_time = rental_time
                customer.rental_basis = 2

        elif choice == "4":
            customer.request_cars()
            rental_time = rental_shop.rent_weekly(customer.cars_rented)
            if rental_time:
                customer.rental_time = rental_time
                customer.rental_basis = 3

        elif choice == "5":
            request = customer.return_cars()
            bill = rental_shop.return_cars(request)
            customer.cars_rented = 0
            customer.rental_basis = 0
            customer.rental_time = None

        elif choice == "6":
            break

        else:
            print("Invalid input. Please enter a number between 1 and 6.")

if __name__ == "__main__":
    main()
```

Code Output:

```
====== Car Rental Shop ======
        1. Display available cars
        2. Rent cars on hourly basis ($5 per hour)
        3. Rent cars on daily basis ($20 per day)
        4. Rent cars on weekly basis ($60 per week)
        5. Return rented cars
        6. Exit

Enter your choice:  1
Available cars: 10

        ====== Car Rental Shop ======
        1. Display available cars
        2. Rent cars on hourly basis ($5 per hour)
        3. Rent cars on daily basis ($20 per day)
        4. Rent cars on weekly basis ($60 per week)
        5. Return rented cars
        6. Exit

Enter your choice:  2
How many cars would you like to rent? 2
2 car(s) rented on hourly basis at 9 hours.

        ====== Car Rental Shop ======
        1. Display available cars
        2. Rent cars on hourly basis ($5 per hour)
        3. Rent cars on daily basis ($20 per day)
        4. Rent cars on weekly basis ($60 per week)
        5. Return rented cars
        6. Exit

Enter your choice:  3
How many cars would you like to rent? 3
3 car(s) rented on daily basis at 7 day.

        ====== Car Rental Shop ======
        1. Display available cars
        2. Rent cars on hourly basis ($5 per hour)
        3. Rent cars on daily basis ($20 per day)
        4. Rent cars on weekly basis ($60 per week)
        5. Return rented cars
        6. Exit

Enter your choice:  4
How many cars would you like to rent? 4
4 car(s) rented on weekly basis at 0 day of the week.

        ====== Car Rental Shop ======
        1. Display available cars
        2. Rent cars on hourly basis ($5 per hour)
        3. Rent cars on daily basis ($20 per day)
        4. Rent cars on weekly basis ($60 per week)
        5. Return rented cars
        6. Exit

Enter your choice:  5
Total bill: $0

        ====== Car Rental Shop ======
        1. Display available cars
        2. Rent cars on hourly basis ($5 per hour)
        3. Rent cars on daily basis ($20 per day)
        4. Rent cars on weekly basis ($60 per week)
        5. Return rented cars
        6. Exit

Enter your choice:  6
```