**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 rentaland import the built-in module DateTimeto 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.

CODE STARTS HERE

# car\_rental.py

import datetime

import random

class CarRental:

def \_\_init\_\_(self, stock=10):

"""Initialize car rental with default or specified stock"""

self.stock = {

'hourly': stock,

'daily': stock,

'weekly': stock

}

self.rental\_history = []

def display\_available\_cars(self):

"""Display available cars for each rental type"""

print("\n--- Available Cars ---")

for rental\_type, count in self.stock.items():

print(f"{rental\_type.capitalize()} Rental: {count} cars available")

def rent\_car(self, request\_type, num\_cars, customer):

"""Rent cars based on rental type"""

if num\_cars <= 0:

print("Number of cars should be positive.")

return False

if num\_cars > self.stock[request\_type]:

print(f"Sorry, only {self.stock[request\_type]} {request\_type} cars available.")

return False

# Update stock and record rental

self.stock[request\_type] -= num\_cars

rental\_record = {

'customer\_id': customer.customer\_id,

'customer\_name': customer.name,

'rental\_type': request\_type,

'num\_cars': num\_cars,

'start\_time': datetime.datetime.now()

}

self.rental\_history.append(rental\_record)

# NEW: Display customer ID when renting

print(f"{num\_cars} {request\_type} car(s) rented successfully.")

print(f"Your Customer ID is: {customer.customer\_id}")

print("Please save this ID for returning the car.")

return True

# Rest of the code remains the same as previous implementation

def return\_car(self, customer, request\_type, num\_cars):

"""Process car return and generate bill"""

# Find the matching rental record

rental\_record = next(

(record for record in self.rental\_history

if record['customer\_id'] == customer.customer\_id and

record['rental\_type'] == request\_type and

record['num\_cars'] == num\_cars),

None

)

if not rental\_record:

print("No matching rental found.")

return False

# Calculate rental duration and bill

end\_time = datetime.datetime.now()

rental\_duration = end\_time - rental\_record['start\_time']

# Rate calculation (in Indian Rupees)

rates = {

'hourly': 100, # ₹100 per hour

'daily': 1000, # ₹1000 per day

'weekly': 5000 # ₹5000 per week

}

if request\_type == 'hourly':

bill = rates['hourly'] \* rental\_duration.total\_seconds() / 3600

elif request\_type == 'daily':

bill = rates['daily'] \* rental\_duration.days

else: # weekly

bill = rates['weekly'] \* (rental\_duration.days // 7)

# Update stock

self.stock[request\_type] += num\_cars

self.rental\_history.remove(rental\_record)

# Print bill

print("\n--- Rental Bill ---")

print(f"Customer: {customer.name}")

print(f"Customer ID: {customer.customer\_id}")

print(f"Rental Type: {request\_type}")

print(f"Number of Cars: {num\_cars}")

print(f"Rental Duration: {rental\_duration}")

print(f"Total Bill: ₹{bill:.2f}")

return True

#Customer Module

import random

#Class Customer

class Customer:

def \_\_init\_\_(self, name):

"""Initialize customer with a unique ID"""

self.name = name

self.customer\_id = self.generate\_customer\_id()

def generate\_customer\_id(self):

"""Generate a unique customer ID"""

return random.randint(1000, 9999)

#Main File

#Complete progam made by Claude

#To be modified into OOP

import car\_rental as cr

import customer as cs

#Main Function

def main():

# Main function implementation stays the same

rental\_system = cr.CarRental()

while True:

print("\n--- Car Rental System ---")

print("1. Display Available Cars")

print("2. Rent a Car")

print("3. Return a Car")

print("4. Exit")

choice = input("Enter your choice (1-4): ")

if choice == '1':

rental\_system.display\_available\_cars()

elif choice == '2':

name = input("Enter your name: ")

customer = cs.Customer(name)

print("\nRental Types:")

print("1. Hourly")

print("2. Daily")

print("3. Weekly")

rental\_type = input("Choose rental type (1-3): ")

rental\_types = ['hourly', 'daily', 'weekly']

num\_cars = int(input("Enter number of cars to rent: "))

rental\_system.rent\_car(

rental\_types[int(rental\_type)-1],

num\_cars,

customer

)

elif choice == '3':

name = input("Enter your name: ")

customer\_id = int(input("Enter your customer ID: "))

print("\nRental Types:")

print("1. Hourly")

print("2. Daily")

print("3. Weekly")

rental\_type = input("Choose rental type (1-3): ")

rental\_types = ['hourly', 'daily', 'weekly']

num\_cars = int(input("Enter number of cars to return: "))

customer = Customer(name)

customer.customer\_id = customer\_id

rental\_system.return\_car(

customer,

rental\_types[int(rental\_type)-1],

num\_cars

)

elif choice == '4':

print("Thank you for using Car Rental System!")

break

else:

print("Invalid choice. Please try again.")

if \_\_name\_\_ == "\_\_main\_\_":

main()