**SimpliLearn**

ARTIFICIAL INTELLIGENCE ENGINEER



**CAR RENTAL PROJECT -OUTPUT SCREENSHOTS**

**SUBMITTED BY**

NAME **:** **SAJESH KUMAR K P**

Email ID **:** [**sajeshkumarkp@gmail.com**](mailto:sajeshkumarkp@gmail.com)

Contents

[1. Project Requirements 3](#_Toc171993283)

[2. Project Output Screenshots 5](#_Toc171993284)

[2.1 Welcome screen 5](#_Toc171993285)

[2.2 Checking available stock of cars 5](#_Toc171993286)

[2.3 Hourly Rental 6](#_Toc171993287)

[2.4 Daily Rental 8](#_Toc171993288)

[2.5 Weekly Rental 11](#_Toc171993289)

[2.6 Exit 14](#_Toc171993290)

[2.7 Validations provided 14](#_Toc171993291)

[3. Summary 17](#_Toc171993292)

# Project Requirements

**Project Name: 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.

# Project Output Screenshots

### Welcome screen

#### Welcome screen

#### 

### Checking available stock of cars

#### Giving input as 0

#### 

#### Showing the available number of cars

#### 

### Hourly Rental

#### Requesting hourly with inputs: 12 cars from 07:00AM

#### 

#### Issuing hourly rental

#### 

#### Checking balance

#### 

#### Showing balance

#### 

#### Returning cars at 4:00PM (16:00)

#### 

#### Returning and generates bill

#### 

### Daily Rental

#### Requesting cars daily with inputs: 14 cars from 01 Jun 2024

#### 

#### Issues car on daily rent

#### 

#### Checking balance

#### 

#### Showing balance

#### 

#### Returning cars on 15 Jun 2024

#### 

#### Returning and generates bill

#### 

### Weekly Rental

#### Requesting cars weekly with inputs: 16 cars for 2 weeks

#### 

#### Issues car on weekly rent

#### 

#### Checking balance

#### 

#### Showing balance

#### 

#### Returning cars (after 2 weeks)

#### 

#### Returning and generates bill

#### 

### Exit

#### Exiting from the system by entering 7

#### 

### Validations provided

#### Invalid main input

#### 

#### Asking more than the stock available: implemented in hourly, daily and weekly showing screenshot from hourly

#### 

#### Date validation

#### 

#### Date difference validation

#### 

# Summary

Submitted all the screenshots of the car rental project