Online Car Rental Platform – Python Project

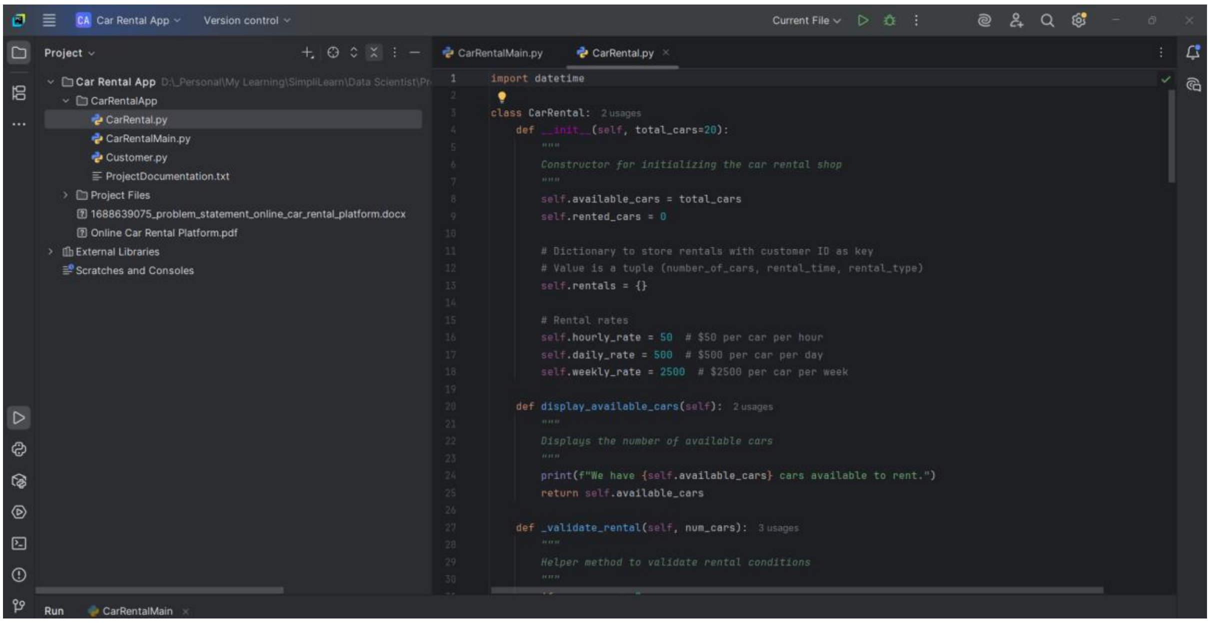
1. Introduction

- Objective of the project (Online Car Rental Platform).
- Tools & Technologies used (Python, Jupyter Notebook, DateTime module, OOP).
- Short summary of the problem statement.

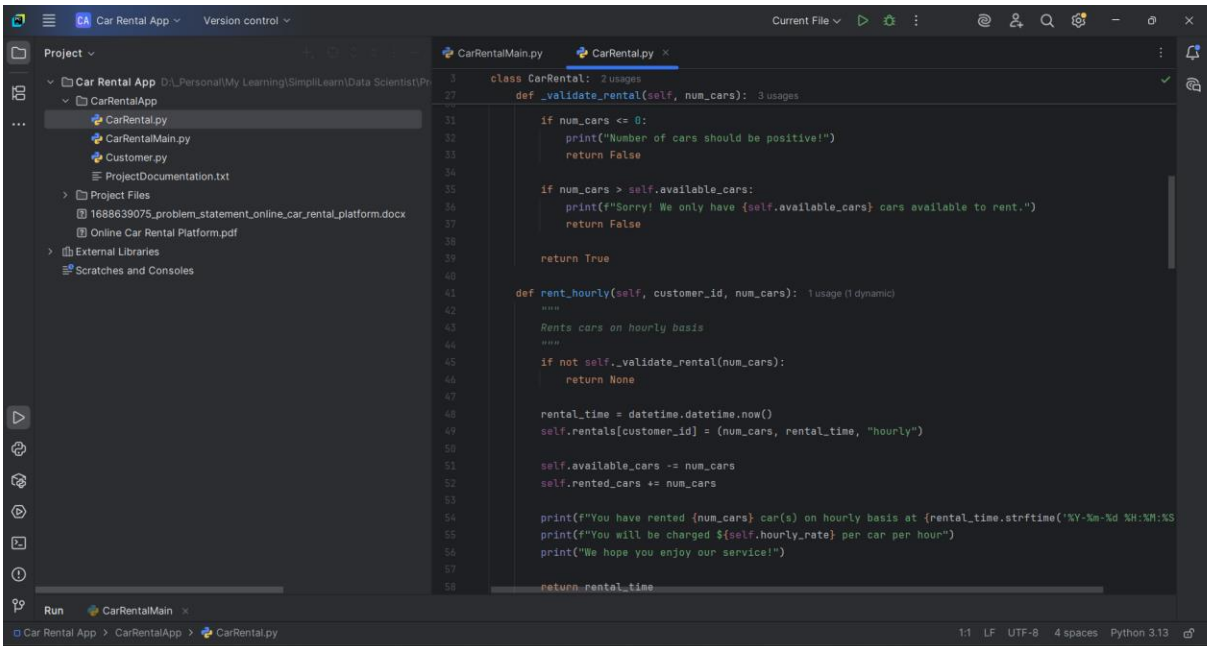
2. Step-by-Step Process

Step 1: Created Car Rental Module

- **Action:** Created a Python module with a CarRental class.
- **Supporting Work:** Defined constructor, initialized available cars.
- **Value:** Encapsulates rental logic for reuse.



```python
import datetime

class CarRental:
    def __init__(self, total_cars=20):
        """
        Constructor for initializing the car rental shop
        """
        self.available_cars = total_cars
        self.rented_cars = 0

        # Dictionary to store rentals with customer ID as key
        # Value is a tuple (number_of_cars, rental_time, rental_type)
        self.rentals = {}

        # Rental rates
        self.hourly_rate = 50  # $50 per car per hour
        self.daily_rate = 500  # $500 per car per day
        self.weekly_rate = 2500  # $2500 per car per week

    def display_available_cars(self):
        """
        Displays the number of available cars
        """
        print(f"We have {self.available_cars} cars available to rent.")
        return self.available_cars

    def _validate_rental(self, num_cars):
        """
        Helper method to validate rental conditions
        """
```

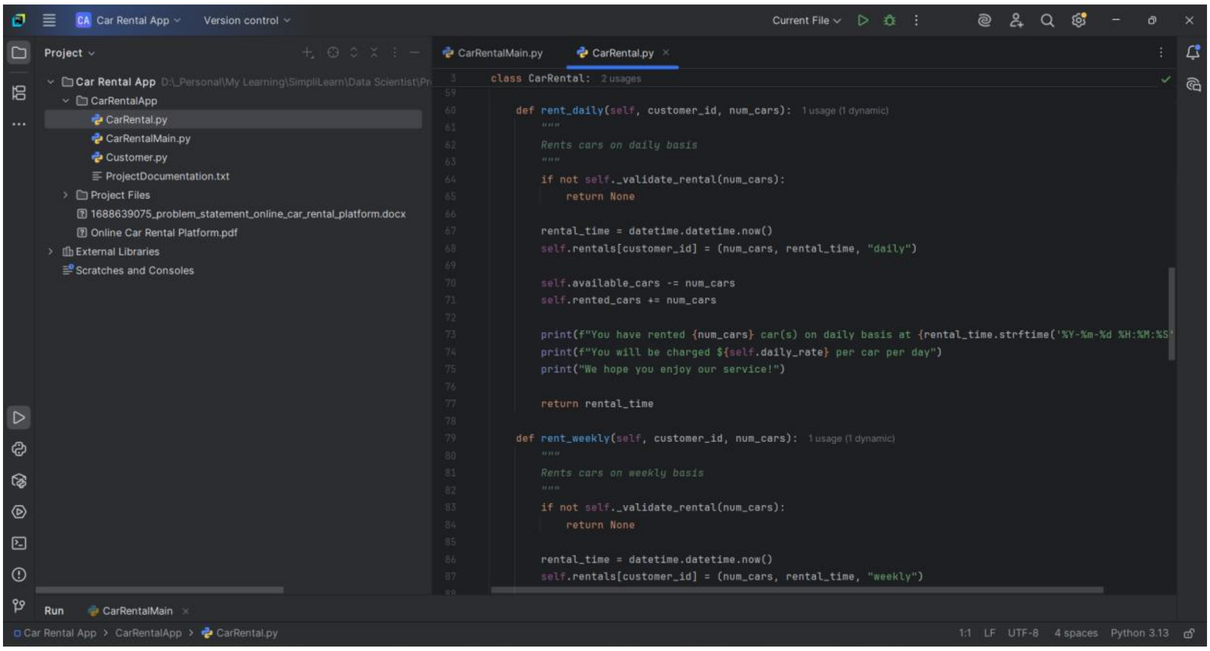


Car Rental App
Version control
Current File
Project
Car Rental App D:_Personal\My Learning\SimpliLearn\Data Scientist\Pr
CarRentalApp
CarRental.py
CarRentalMain.py
Customer.py
ProjectDocumentation.txt
Project Files
1688639075_problem_statement_online_car_rental_platform.docx
Online Car Rental Platform.pdf
External Libraries
Scratches and Consoles
CarRentalMain.py
CarRental.py
Run
CarRentalMain
Car Rental App > CarRentalApp > CarRental.py
1:1 LF UTF-8 4 spaces Python 3.13

```python
class CarRental:  2 usages
    def _validate_rental(self, num_cars):  3 usages
        if num_cars <= 0:
            print("Number of cars should be positive!")
            return False

        if num_cars > self.available_cars:
            print(f"Sorry! We only have {self.available_cars} cars available to rent.")
            return False

        return True

    def rent_hourly(self, customer_id, num_cars):  1 usage (1 dynamic)
        """
        Rents cars on hourly basis
        """
        if not self._validate_rental(num_cars):
            return None

        rental_time = datetime.datetime.now()
        self.rentals[customer_id] = (num_cars, rental_time, "hourly")

        self.available_cars -= num_cars
        self.rented_cars += num_cars

        print(f"You have rented {num_cars} car(s) on hourly basis at {rental_time.strftime('%Y-%m-%d %H:%M:%S
        print(f"You will be charged ${self.hourly_rate} per car per hour")
        print("We hope you enjoy our service!")

        return rental_time
```

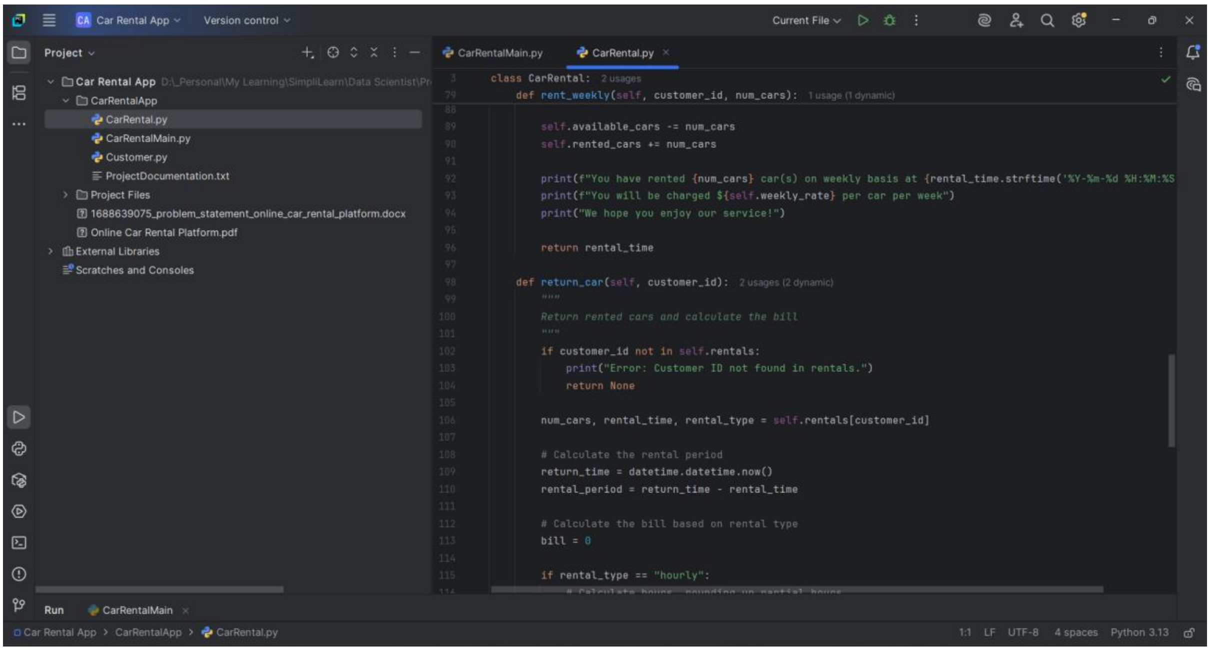
Step 2: Defined Methods for Rental Options

- **Action:** Added methods for hourly, daily, and weekly rentals.
- **Supporting Work:** Validation checks (positive numbers, stock availability).
- **Value:** Provides flexibility in rental modes.



Car Rental App
Version control
Current File
Project
Car Rental App D:_Personal\My Learning\SimpliLearn\Data Scientist\Pr
CarRentalApp
CarRental.py
CarRentalMain.py
Customer.py
ProjectDocumentation.txt
Project Files
1688639075_problem_statement_online_car_rental_platform.docx
Online Car Rental Platform.pdf
External Libraries
Scratches and Consoles
CarRentalMain.py
CarRental.py
Run
CarRentalMain
Car Rental App > CarRentalApp > CarRental.py
1:1 LF UTF-8 4 spaces Python 3.13

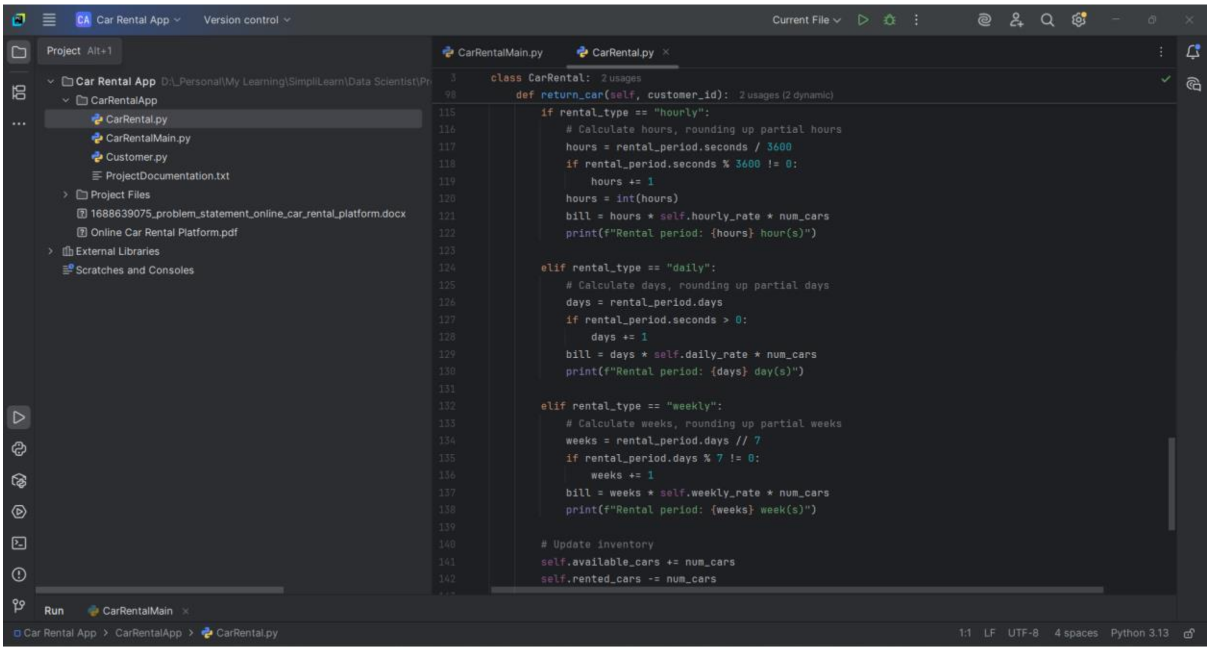
```python
class CarRental:  2 usages

    def rent_daily(self, customer_id, num_cars):  1 usage (1 dynamic)
        """
        Rents cars on daily basis
        """
        if not self._validate_rental(num_cars):
            return None

        rental_time = datetime.datetime.now()
        self.rentals[customer_id] = (num_cars, rental_time, "daily")

        self.available_cars -= num_cars
        self.rented_cars += num_cars

        print(f"You have rented {num_cars} car(s) on daily basis at {rental_time.strftime('%Y-%m-%d %H:%M:%S'
        print(f"You will be charged ${self.daily_rate} per car per day")
        print("We hope you enjoy our service!")

        return rental_time

    def rent_weekly(self, customer_id, num_cars):  1 usage (1 dynamic)
        """
        Rents cars on weekly basis
        """
        if not self._validate_rental(num_cars):
            return None

        rental_time = datetime.datetime.now()
        self.rentals[customer_id] = (num_cars, rental_time, "weekly")
```



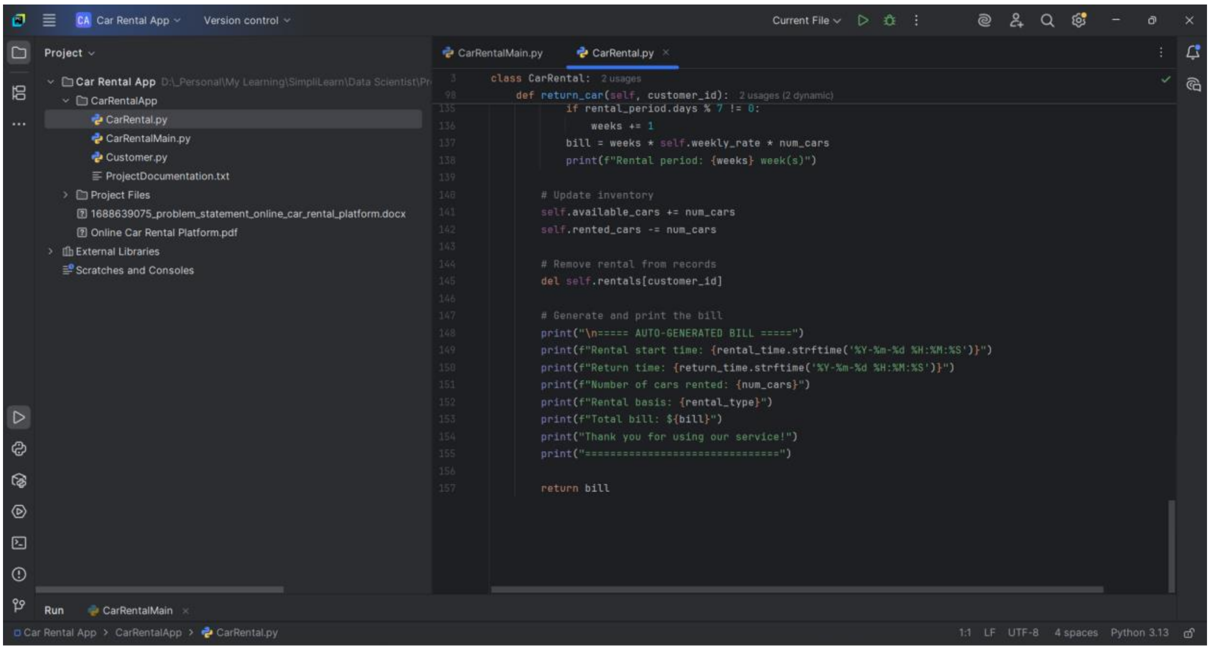
```python
class CarRental:
    def rent_weekly(self, customer_id, num_cars):

            self.available_cars -= num_cars
            self.rented_cars += num_cars

            print(f"You have rented {num_cars} car(s) on weekly basis at {rental_time.strftime('%Y-%m-%d %H:%M:%S
            print(f"You will be charged ${self.weekly_rate} per car per week")
            print("We hope you enjoy our service!")

            return rental_time

    def return_car(self, customer_id):
        """
        Return rented cars and calculate the bill
        """
        if customer_id not in self.rentals:
            print("Error: Customer ID not found in rentals.")
            return None

        num_cars, rental_time, rental_type = self.rentals[customer_id]

        # Calculate the rental period
        return_time = datetime.datetime.now()
        rental_period = return_time - rental_time

        # Calculate the bill based on rental type
        bill = 0

        if rental_type == "hourly":
```

Step 3: Managed Rental Time and Billing

- **Action:** Stored rental start time using datetime.
- **Supporting Work:** Implemented billing logic in return_car() method.
- **Value:** Automatically calculates charges based on rental period and mode.



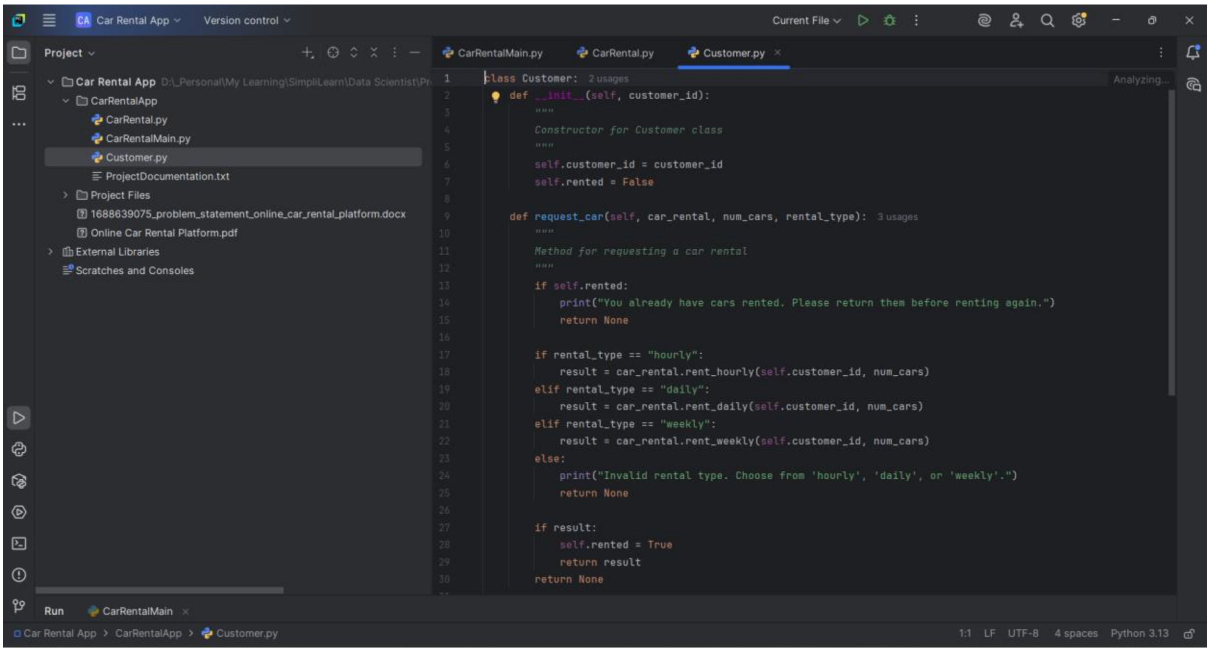
```python
class CarRental:
    def return_car(self, customer_id):
        if rental_type == "hourly":
            # Calculate hours, rounding up partial hours
            hours = rental_period.seconds / 3600
            if rental_period.seconds % 3600 != 0:
                hours += 1
            hours = int(hours)
            bill = hours * self.hourly_rate * num_cars
            print(f"Rental period: {hours} hour(s)")

        elif rental_type == "daily":
            # Calculate days, rounding up partial days
            days = rental_period.days
            if rental_period.seconds > 0:
                days += 1
            bill = days * self.daily_rate * num_cars
            print(f"Rental period: {days} day(s)")

        elif rental_type == "weekly":
            # Calculate weeks, rounding up partial weeks
            weeks = rental_period.days // 7
            if rental_period.days % 7 != 0:
                weeks += 1
            bill = weeks * self.weekly_rate * num_cars
            print(f"Rental period: {weeks} week(s)")

        # Update inventory
        self.available_cars += num_cars
        self.rented_cars -= num_cars
```


Car Rental App
Version control
Current File
Project
Car Rental App D:_Personal\My Learning\SimpliLearn\Data Scientist\Pr
CarRentalApp
CarRental.py
CarRentalMain.py
Customer.py
ProjectDocumentation.txt
Project Files
1688639075_problem_statement_online_car_rental_platform.docx
Online Car Rental Platform.pdf
External Libraries
Scratches and Consoles
CarRentalMain.py
CarRental.py
Run
CarRentalMain
Car Rental App > CarRentalApp > CarRental.py
1:1 LF UTF-8 4 spaces Python 3.13

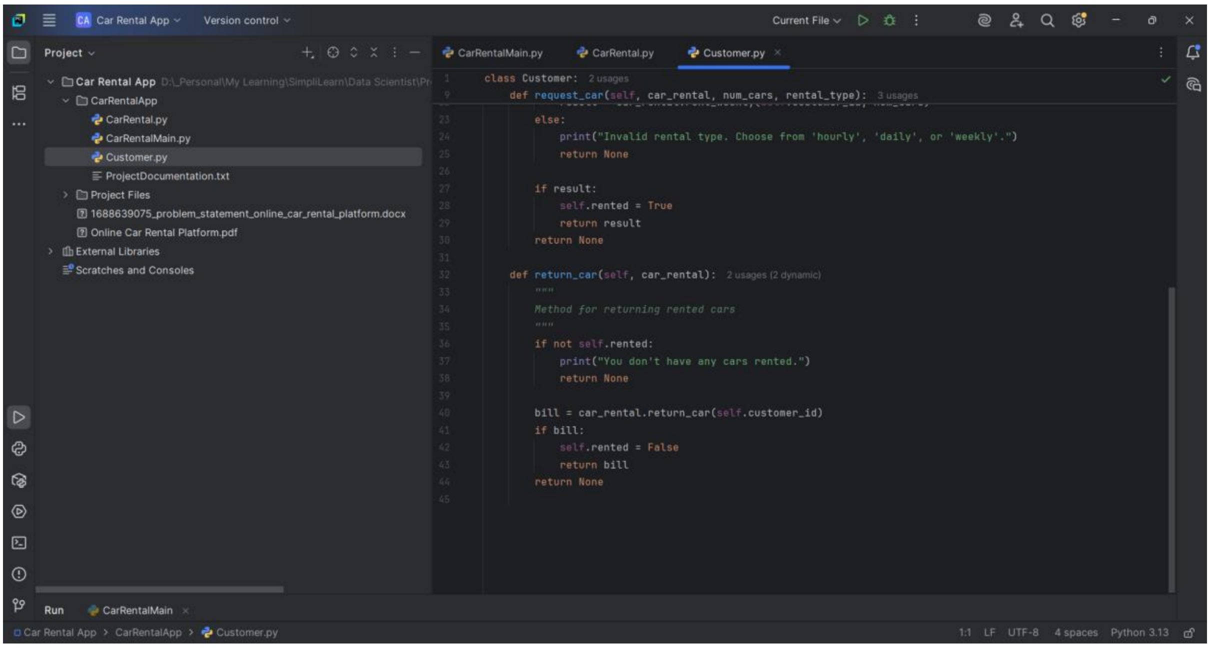
```python
class CarRental:  2 usages
    def return_car(self, customer_id):  2 usages (2 dynamic)
            if rental_period.days % 7 != 0:
                weeks += 1
            bill = weeks * self.weekly_rate * num_cars
            print(f"Rental period: {weeks} week(s)")

        # Update inventory
        self.available_cars += num_cars
        self.rented_cars -= num_cars

        # Remove rental from records
        del self.rentals[customer_id]

        # Generate and print the bill
        print("\n===== AUTO-GENERATED BILL =====")
        print(f"Rental start time: {rental_time.strftime('%Y-%m-%d %H:%M:%S')}")
        print(f"Return time: {return_time.strftime('%Y-%m-%d %H:%M:%S')}")
        print(f"Number of cars rented: {num_cars}")
        print(f"Rental basis: {rental_type}")
        print(f"Total bill: ${bill}")
        print("Thank you for using our service!")
        print("================================")

        return bill
```

Step 4: Created Customer Class

- **Action:** Defined methods for requesting and returning cars.
- **Supporting Work:** Integrated with CarRental methods.
- **Value:** Provides abstraction for customer interactions.



```python
class Customer:  2 usages
    def __init__(self, customer_id):
        """
        Constructor for Customer class
        """
        self.customer_id = customer_id
        self.rented = False

    def request_car(self, car_rental, num_cars, rental_type):  3 usages
        """
        Method for requesting a car rental
        """
        if self.rented:
            print("You already have cars rented. Please return them before renting again.")
            return None

        if rental_type == "hourly":
            result = car_rental.rent_hourly(self.customer_id, num_cars)
        elif rental_type == "daily":
            result = car_rental.rent_daily(self.customer_id, num_cars)
        elif rental_type == "weekly":
            result = car_rental.rent_weekly(self.customer_id, num_cars)
        else:
            print("Invalid rental type. Choose from 'hourly', 'daily', or 'weekly'.")
            return None

        if result:
            self.rented = True
            return result
        return None
```

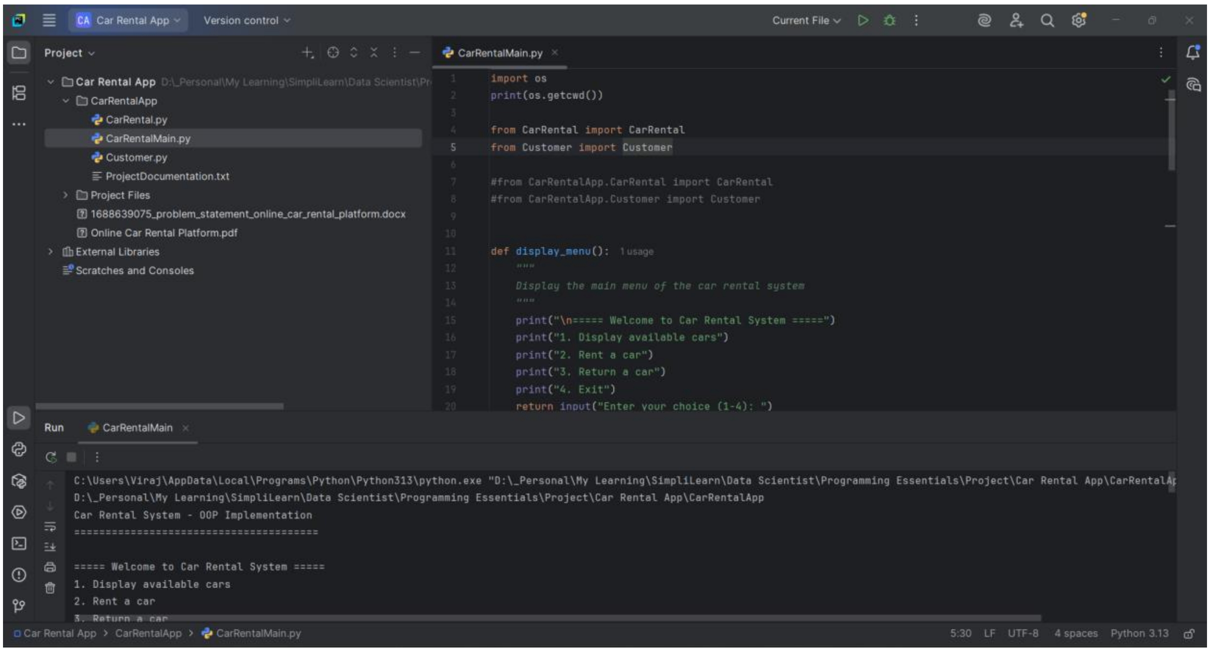


Car Rental App
Version control
Current File
Project
Car Rental App D:_Personal\My Learning\SimpliLearn\Data Scientist\Pr
CarRentalApp
CarRental.py
CarRentalMain.py
Customer.py
ProjectDocumentation.txt
Project Files
1688639075_problem_statement_online_car_rental_platform.docx
Online Car Rental Platform.pdf
External Libraries
Scratches and Consoles
CarRentalMain.py
CarRental.py
Customer.py
Run
CarRentalMain
Car Rental App > CarRentalApp > Customer.py
1:1 LF UTF-8 4 spaces Python 3.13

```
class Customer:  2 usages
    def request_car(self, car_rental, num_cars, rental_type):  3 usages
        else:
            print("Invalid rental type. Choose from 'hourly', 'daily', or 'weekly'.")
            return None

        if result:
            self.rented = True
            return result
        return None

    def return_car(self, car_rental):  2 usages (2 dynamic)
        """
        Method for returning rented cars
        """
        if not self.rented:
            print("You don't have any cars rented.")
            return None

        bill = car_rental.return_car(self.customer_id)
        if bill:
            self.rented = False
            return bill
        return None
```

Step 5: Built Main Project File

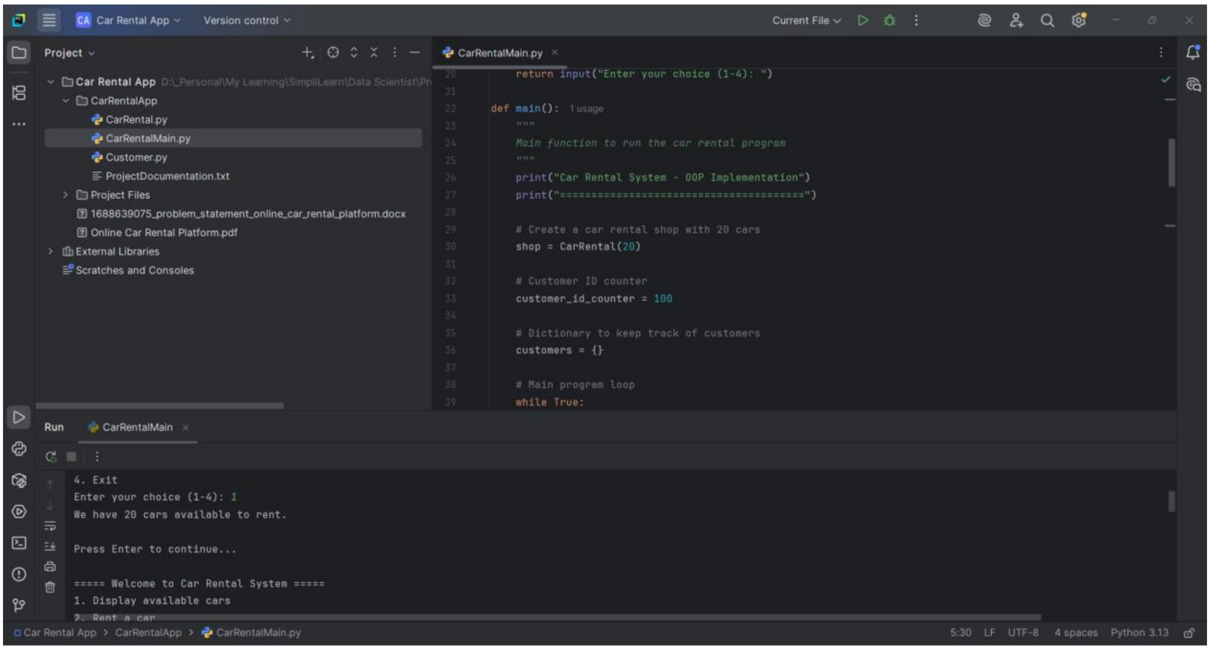
- **Action:** Created .ipynb main file and imported the module.
- **Supporting Work:** Defined a menu-driven system (display cars, rent, return).
- **Value:** User-friendly flow for testing and execution.



Car Rental App
Version control
Current File
Project
Car Rental App D:_Personal\My Learning\SimpliLearn\Data Scientist\Pr
CarRentalApp
CarRental.py
CarRentalMain.py
Customer.py
ProjectDocumentation.txt
Project Files
1688639075_problem_statement_online_car_rental_platform.docx
Online Car Rental Platform.pdf
External Libraries
Scratches and Consoles
CarRentalMain.py
Run
CarRentalMain
Car Rental App > CarRentalApp > CarRentalMain.py
5:30 LF UTF-8 4 spaces Python 3.13

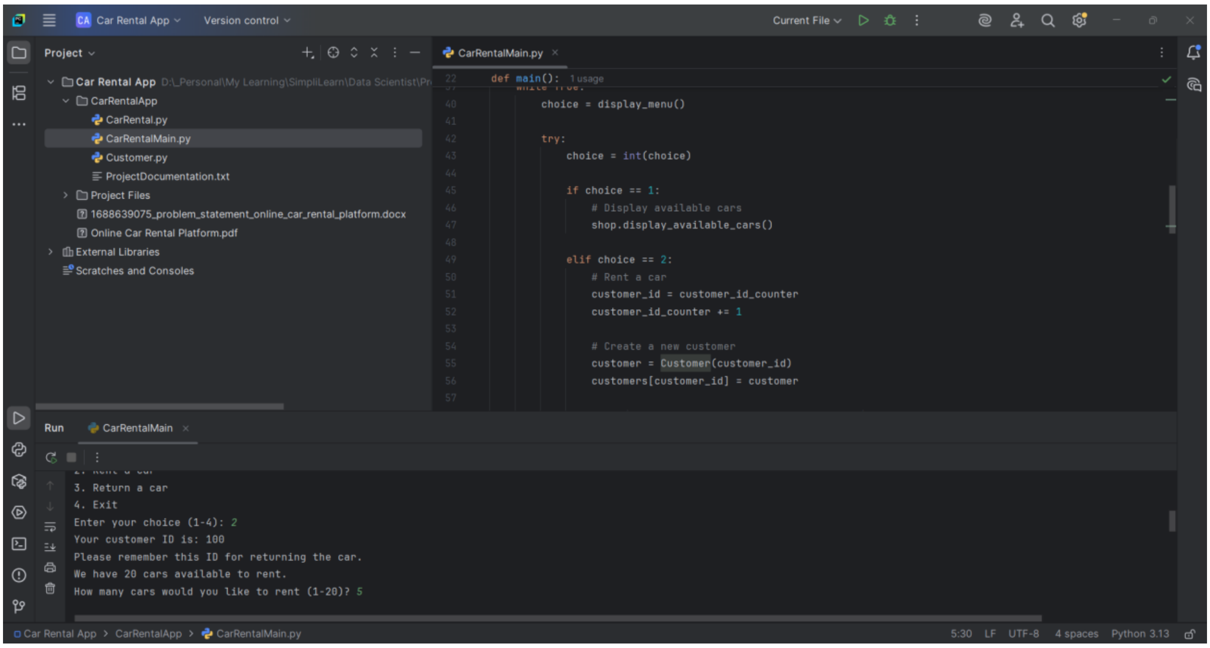
```
import os
print(os.getcwd())

from CarRental import CarRental
from Customer import Customer

#from CarRentalApp.CarRental import CarRental
#from CarRentalApp.Customer import Customer


def display_menu():  1 usage
    """
    Display the main menu of the car rental system
    """
    print("\n===== Welcome to Car Rental System =====")
    print("1. Display available cars")
    print("2. Rent a car")
    print("3. Return a car")
    print("4. Exit")
    return input("Enter your choice (1-4): ")
```

```
C:\Users\Viraj\AppData\Local\Programs\Python\Python313\python.exe "D:\_Personal\My Learning\SimpliLearn\Data Scientist\Programming Essentials\Project\Car Rental App\CarRentalAp
D:\_Personal\My Learning\SimpliLearn\Data Scientist\Programming Essentials\Project\Car Rental App\CarRentalApp
Car Rental System - OOP Implementation
=======================================

===== Welcome to Car Rental System =====
1. Display available cars
2. Rent a car
3. Return a car
```


Car Rental App
Version control
Current File
Project
Car Rental App D:_Personal\My Learning\SimpliLearn\Data Scientist\Pr
CarRentalApp
CarRental.py
CarRentalMain.py
Customer.py
ProjectDocumentation.txt
Project Files
1688639075_problem_statement_online_car_rental_platform.docx
Online Car Rental Platform.pdf
External Libraries
Scratches and Consoles
CarRentalMain.py
Run
CarRentalMain
Car Rental App > CarRentalApp > CarRentalMain.py
5:30 LF UTF-8 4 spaces Python 3.13

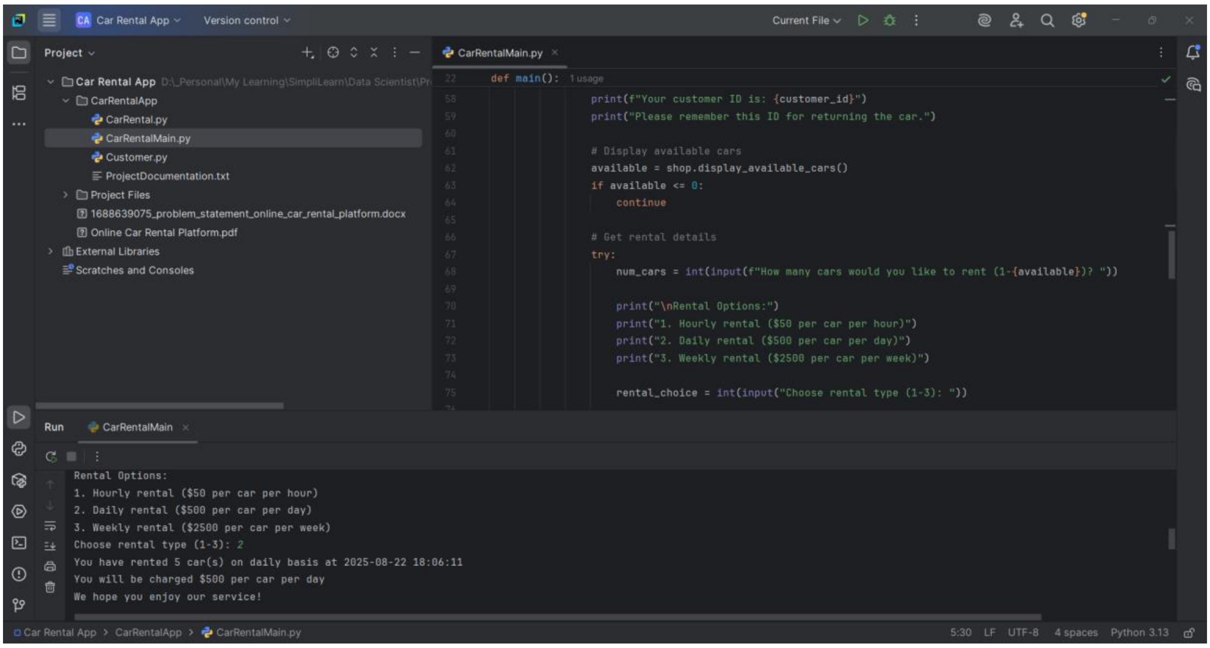
```python
        return input("Enter your choice (1-4): ")

def main():  1 usage
    """
    Main function to run the car rental program
    """
    print("Car Rental System - OOP Implementation")
    print("=====================================")

    # Create a car rental shop with 20 cars
    shop = CarRental(20)

    # Customer ID counter
    customer_id_counter = 100

    # Dictionary to keep track of customers
    customers = {}

    # Main program loop
    while True:
```

```
4. Exit
Enter your choice (1-4): 1
We have 20 cars available to rent.

Press Enter to continue...

===== Welcome to Car Rental System =====
1. Display available cars
2. Rent a car
```


Car Rental App
Version control
Current File
Project
Car Rental App D:_Personal\My Learning\SimpliLearn\Data Scientist\Pr
CarRentalApp
CarRental.py
CarRentalMain.py
Customer.py
ProjectDocumentation.txt
Project Files
1688639075_problem_statement_online_car_rental_platform.docx
Online Car Rental Platform.pdf
External Libraries
Scratches and Consoles
CarRentalMain.py
Run
CarRentalMain
Car Rental App > CarRentalApp > CarRentalMain.py
5:30 LF UTF-8 4 spaces Python 3.13

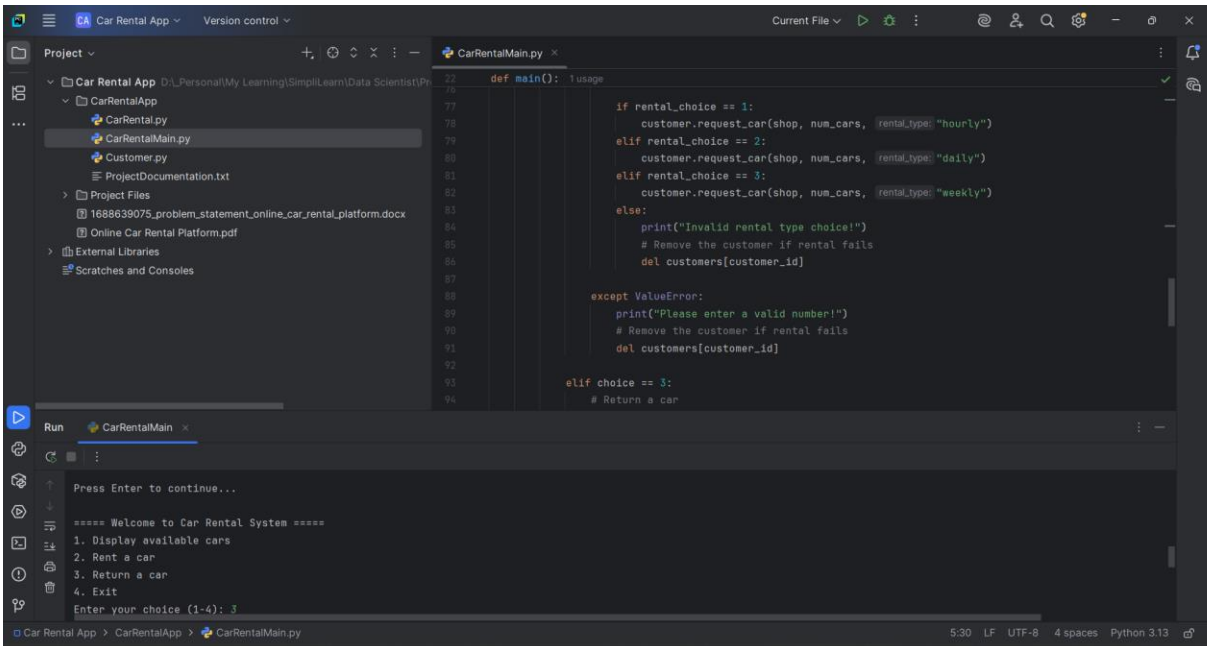
```python
def main():  1 usage
        choice = display_menu()

        try:
            choice = int(choice)

            if choice == 1:
                # Display available cars
                shop.display_available_cars()

            elif choice == 2:
                # Rent a car
                customer_id = customer_id_counter
                customer_id_counter += 1

                # Create a new customer
                customer = Customer(customer_id)
                customers[customer_id] = customer

```

```
2. Rent a car
3. Return a car
4. Exit
Enter your choice (1-4): 2
Your customer ID is: 100
Please remember this ID for returning the car.
We have 20 cars available to rent.
How many cars would you like to rent (1-20)? 5
```


Car Rental App
Version control
Current File
Project
Car Rental App D:_Personal\My Learning\SimpliLearn\Data Scientist\Pr
CarRentalApp
CarRental.py
CarRentalMain.py
Customer.py
ProjectDocumentation.txt
Project Files
1688639075_problem_statement_online_car_rental_platform.docx
Online Car Rental Platform.pdf
External Libraries
Scratches and Consoles
CarRentalMain.py
Run
CarRentalMain
Car Rental App > CarRentalApp > CarRentalMain.py
5:30 LF UTF-8 4 spaces Python 3.13

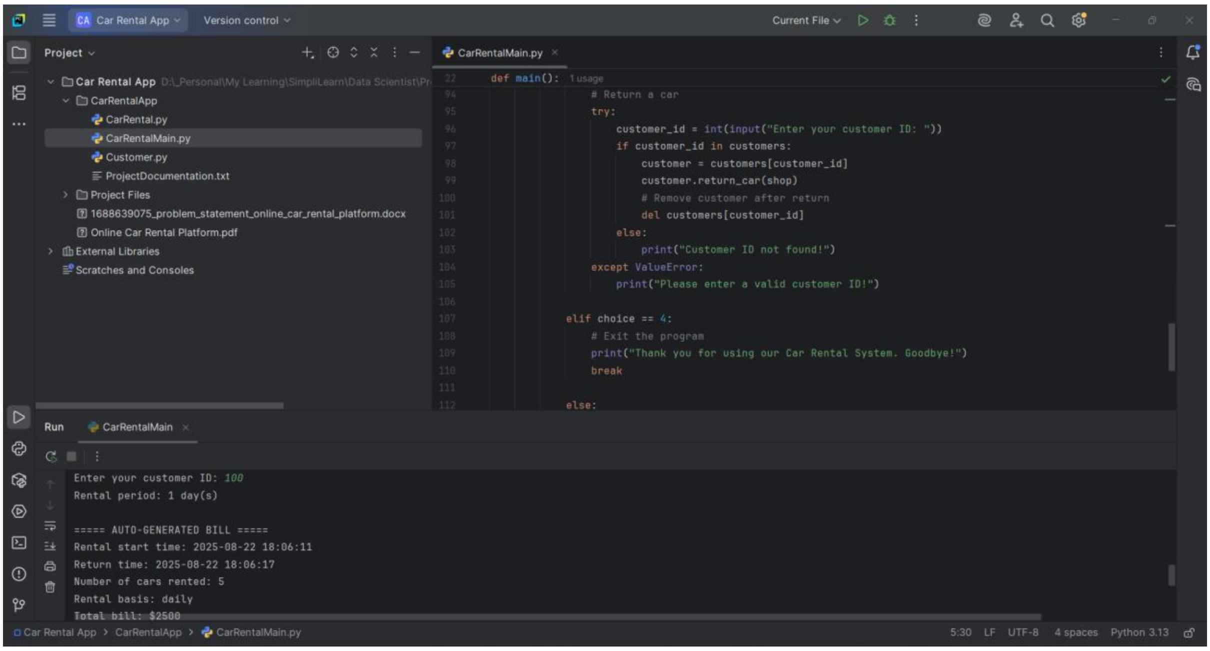
```python
def main():  1 usage
                print(f"Your customer ID is: {customer_id}")
                print("Please remember this ID for returning the car.")

                # Display available cars
                available = shop.display_available_cars()
                if available <= 0:
                    continue

                # Get rental details
                try:
                    num_cars = int(input(f"How many cars would you like to rent (1-{available})? "))

                    print("\nRental Options:")
                    print("1. Hourly rental ($50 per car per hour)")
                    print("2. Daily rental ($500 per car per day)")
                    print("3. Weekly rental ($2500 per car per week)")

                    rental_choice = int(input("Choose rental type (1-3): "))
```

```
Rental Options:
1. Hourly rental ($50 per car per hour)
2. Daily rental ($500 per car per day)
3. Weekly rental ($2500 per car per week)
Choose rental type (1-3): 2
You have rented 5 car(s) on daily basis at 2025-08-22 18:06:11
You will be charged $500 per car per day
We hope you enjoy our service!
```


Car Rental App
Version control
Current File
Project
Car Rental App D:_Personal\My Learning\SimpliLearn\Data Scientist\Pr
CarRentalApp
CarRental.py
CarRentalMain.py
Customer.py
ProjectDocumentation.txt
Project Files
1688639075_problem_statement_online_car_rental_platform.docx
Online Car Rental Platform.pdf
External Libraries
Scratches and Consoles
CarRentalMain.py
Run
CarRentalMain
Car Rental App > CarRentalApp > CarRentalMain.py
5:30 LF UTF-8 4 spaces Python 3.13

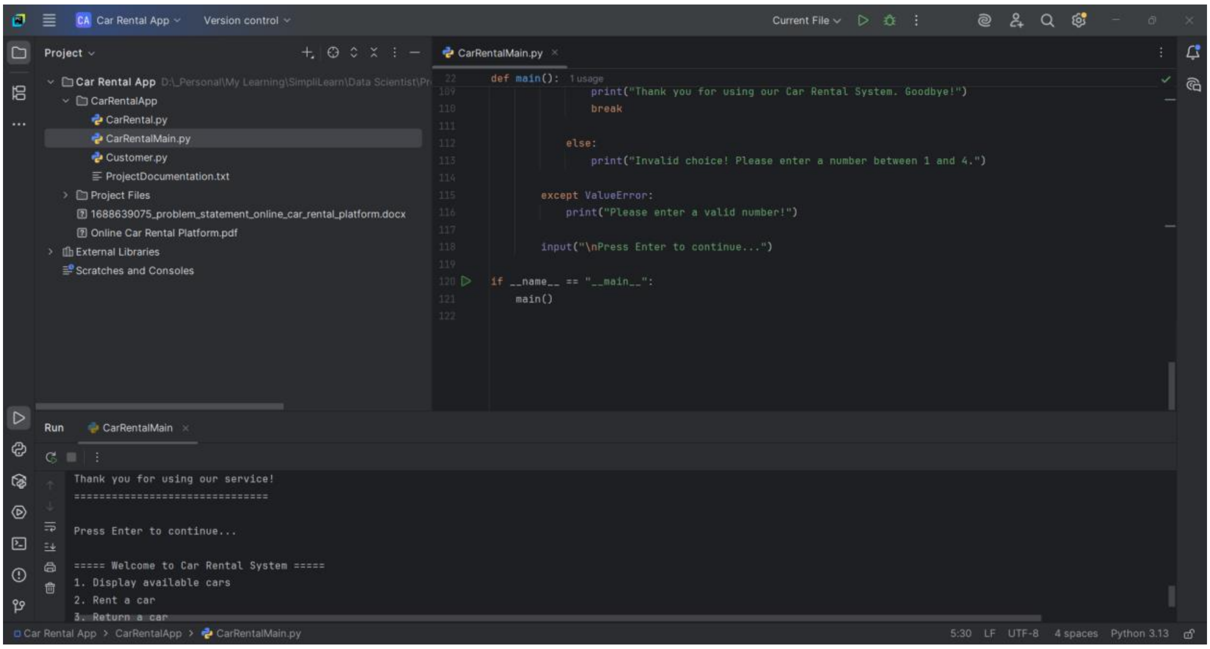
```python
def main():  1 usage

                    if rental_choice == 1:
                        customer.request_car(shop, num_cars, rental_type: "hourly")
                    elif rental_choice == 2:
                        customer.request_car(shop, num_cars, rental_type: "daily")
                    elif rental_choice == 3:
                        customer.request_car(shop, num_cars, rental_type: "weekly")
                    else:
                        print("Invalid rental type choice!")
                        # Remove the customer if rental fails
                        del customers[customer_id]

                except ValueError:
                    print("Please enter a valid number!")
                    # Remove the customer if rental fails
                    del customers[customer_id]

            elif choice == 3:
                # Return a car
```

```
Press Enter to continue...

===== Welcome to Car Rental System =====
1. Display available cars
2. Rent a car
3. Return a car
4. Exit
Enter your choice (1-4): 3
```


Car Rental App
Version control
Current File
Project
Car Rental App D:_Personal\My Learning\SimpliLearn\Data Scientist\Pr
CarRentalApp
CarRental.py
CarRentalMain.py
Customer.py
ProjectDocumentation.txt
Project Files
1688639075_problem_statement_online_car_rental_platform.docx
Online Car Rental Platform.pdf
External Libraries
Scratches and Consoles
CarRentalMain.py
1 usage
Run
CarRentalMain
Car Rental App > CarRentalApp > CarRentalMain.py
5:30 LF UTF-8 4 spaces Python 3.13

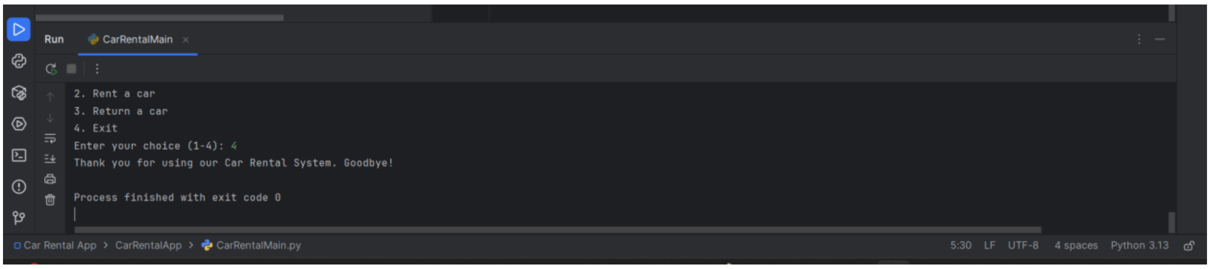
```python
def main():
                # Return a car
                try:
                    customer_id = int(input("Enter your customer ID: "))
                    if customer_id in customers:
                        customer = customers[customer_id]
                        customer.return_car(shop)
                        # Remove customer after return
                        del customers[customer_id]
                    else:
                        print("Customer ID not found!")
                except ValueError:
                    print("Please enter a valid customer ID!")

            elif choice == 4:
                # Exit the program
                print("Thank you for using our Car Rental System. Goodbye!")
                break

            else:
```

```
Enter your customer ID: 100
Rental period: 1 day(s)

===== AUTO-GENERATED BILL =====
Rental start time: 2025-08-22 18:06:11
Return time: 2025-08-22 18:06:17
Number of cars rented: 5
Rental basis: daily
Total bill: $2500
```


Car Rental App
Version control
Current File
Project
Car Rental App D:_Personal\My Learning\SimpliLearn\Data Scientist\Pr
CarRentalApp
CarRental.py
CarRentalMain.py
Customer.py
ProjectDocumentation.txt
Project Files
1688639075_problem_statement_online_car_rental_platform.docx
Online Car Rental Platform.pdf
External Libraries
Scratches and Consoles
CarRentalMain.py
1 usage
Run
CarRentalMain
Car Rental App > CarRentalApp > CarRentalMain.py
5:30 LF UTF-8 4 spaces Python 3.13

```python
def main():
                print("Thank you for using our Car Rental System. Goodbye!")
                break

            else:
                print("Invalid choice! Please enter a number between 1 and 4.")

        except ValueError:
            print("Please enter a valid number!")

        input("\nPress Enter to continue...")

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

```
Thank you for using our service!
==============================

Press Enter to continue...

===== Welcome to Car Rental System =====
1. Display available cars
2. Rent a car
3. Return a car
```


Run
CarRentalMain
Car Rental App > CarRentalApp > CarRentalMain.py
5:30 LF UTF-8 4 spaces Python 3.13

```
2. Rent a car
3. Return a car
4. Exit
Enter your choice (1-4): 4
Thank you for using our Car Rental System. Goodbye!

Process finished with exit code 0
```

Step 6: Tested the Project

- **Action:** Executed test cases for hourly/daily/weekly rentals.
- **Supporting Work:** Verified inventory updates and bill correctness.
- **Value:** Confirms correctness and robustness.

3. Output & Results

- **Screenshots:** Renting cars, returning cars, and sample bills (Above Screenshots for reference).
- **Key Observations:**
  - Inventory updates after each rental/return.
  - Bill shows rental period, number of cars, and total cost.
  - Proper error handling when invalid inputs are given.

4. Conclusion

- The project successfully simulates a **real-world car rental system**.
- Showcases **Object-Oriented Programming concepts**: classes, methods, objects, encapsulation.
- Practical learning in **DateTime handling, validation, and user interaction**.
- Prepares for **future projects/interviews** by demonstrating ability to design structured, reusable code.