

Car Rental System

This is a simple console-based car rental system implemented in C++. The system allows users to display available cars, rent a car, and exit the program.

Structures and Functions.

Car Structure

The **Car** structure represents a car with the following attributes:

- **company**: a string representing the company of the car
- **model**: a string representing the model of the car
- **year**: an integer representing the year of the car
- **rgno**: an integer representing the registration number of the car
- **rent**: an integer representing the daily rent of the car
- **rented**: a boolean indicating whether the car is currently rented or not

displayCar Function

- The **displayCar** function takes a **const Car&** object as an argument and displays the details of the car to the console.

rentCar Function

- The **rentCar** function takes a **Car&** object as an argument and allows the user to rent the car if it is not already rented. If the car is already rented, it displays an error message.

displayMenu Function

- The **displayMenu** function displays the main menu of the car rental system to the console.

Main Program

- The main program creates three **Car** objects: **car123**, **car345**, and **car567**. It then enters an infinite loop, displaying the main menu to the user and processing their input.

Menu Options

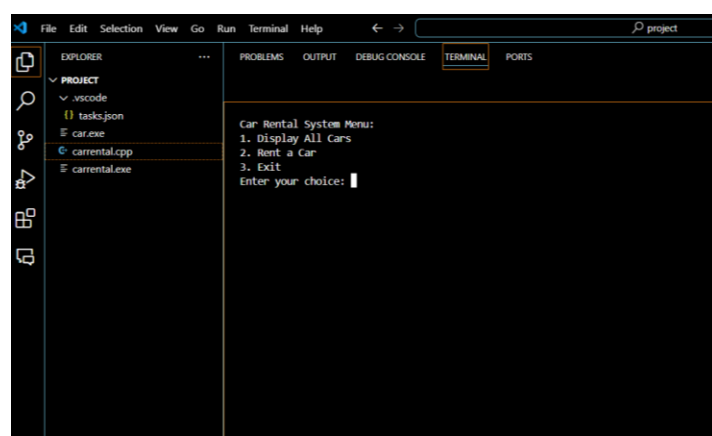
The main menu has three options:

- **1. Display All Cars**: Displays the details of all available cars.
- **2. Rent a Car**: Allows the user to rent a car by entering its registration number.
- **3. Exit**: Exits the program.

Renting a Car:

- When the user chooses to rent a car, they are prompted to enter the registration number of the car they want to rent. The program then checks if the car is available and allows the user to rent it if it is not already rented.

Sample Output:



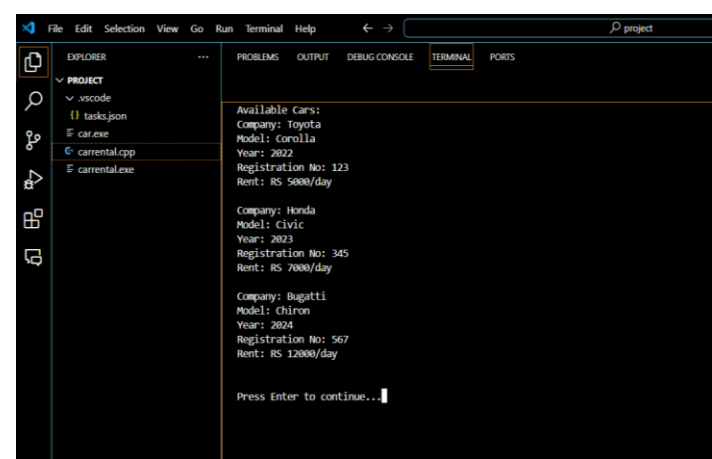
```
File Edit Selection View Go Run Terminal Help
PROJECT
  .vscode
  tasks.json
  car.exe
  carrental.cpp
  carrental.exe

Car Rental System Menu:
1. Display All Cars
2. Rent a Car
3. Exit
Enter your choice: |
```



```
File Edit Selection View Go Run Terminal Help
PROJECT
  .vscode
  tasks.json
  car.exe
  carrental.cpp
  carrental.exe

Enter the Registration Number of the car you want to rent: |
```



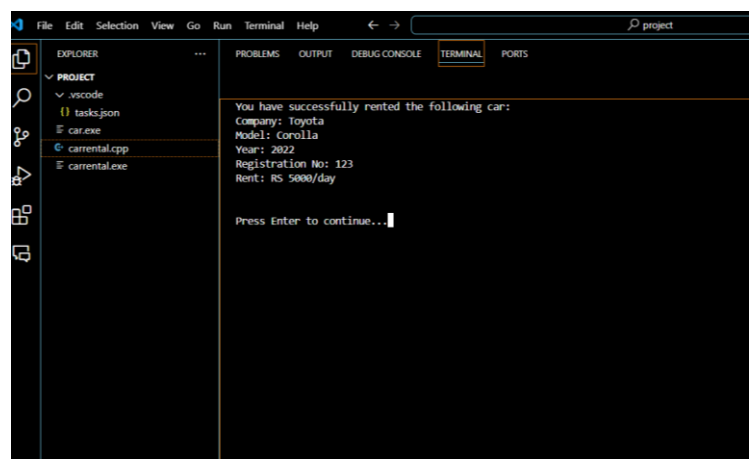
```
File Edit Selection View Go Run Terminal Help
PROJECT
  .vscode
  tasks.json
  car.exe
  carrental.cpp
  carrental.exe

Available Cars:
Company: Toyota
Model: Corolla
Year: 2022
Registration No: 123
Rent: RS 5000/day

Company: Honda
Model: Civic
Year: 2023
Registration No: 345
Rent: RS 7000/day

Company: Bugatti
Model: Chiron
Year: 2024
Registration No: 567
Rent: RS 12000/day

Press Enter to continue...|
```



```
File Edit Selection View Go Run Terminal Help
PROJECT
  .vscode
  tasks.json
  car.exe
  carrental.cpp
  carrental.exe

You have successfully rented the following car:
Company: Toyota
Model: Corolla
Year: 2022
Registration No: 123
Rent: RS 5000/day

Press Enter to continue...|
```

Notes

- The program uses the **system("clear")** and **system("cls")** functions to clear the console screen, which may not work on all platforms.
- The program uses **cin.ignore()** and **cin.get()** to pause the program and wait for the user to press Enter before continuing.

Conclusion:

The Car Rental System effectively demonstrates the fundamental concepts of C++ programming, including structures, functions, and basic user interaction. It provides a clear and functional interface for displaying available cars, handling car rentals, and exiting the application. The use of the `Car` structure to manage car details and rental status simplifies code organization and readability. While the system meets its primary objectives, improvements such as enhanced error handling and cross-platform compatibility could further enhance its functionality and user experience. Overall, the system serves as a practical and educational example of managing simple rental operations in C++.