

Parking Lot

Overview: This repository gives an overview of how we can design a basic parking lot in Python. It creates parking lot with given number of slots. The cars follow Greedy approach while being parked in the slots.

File Structure: I have divided my project into four modules:

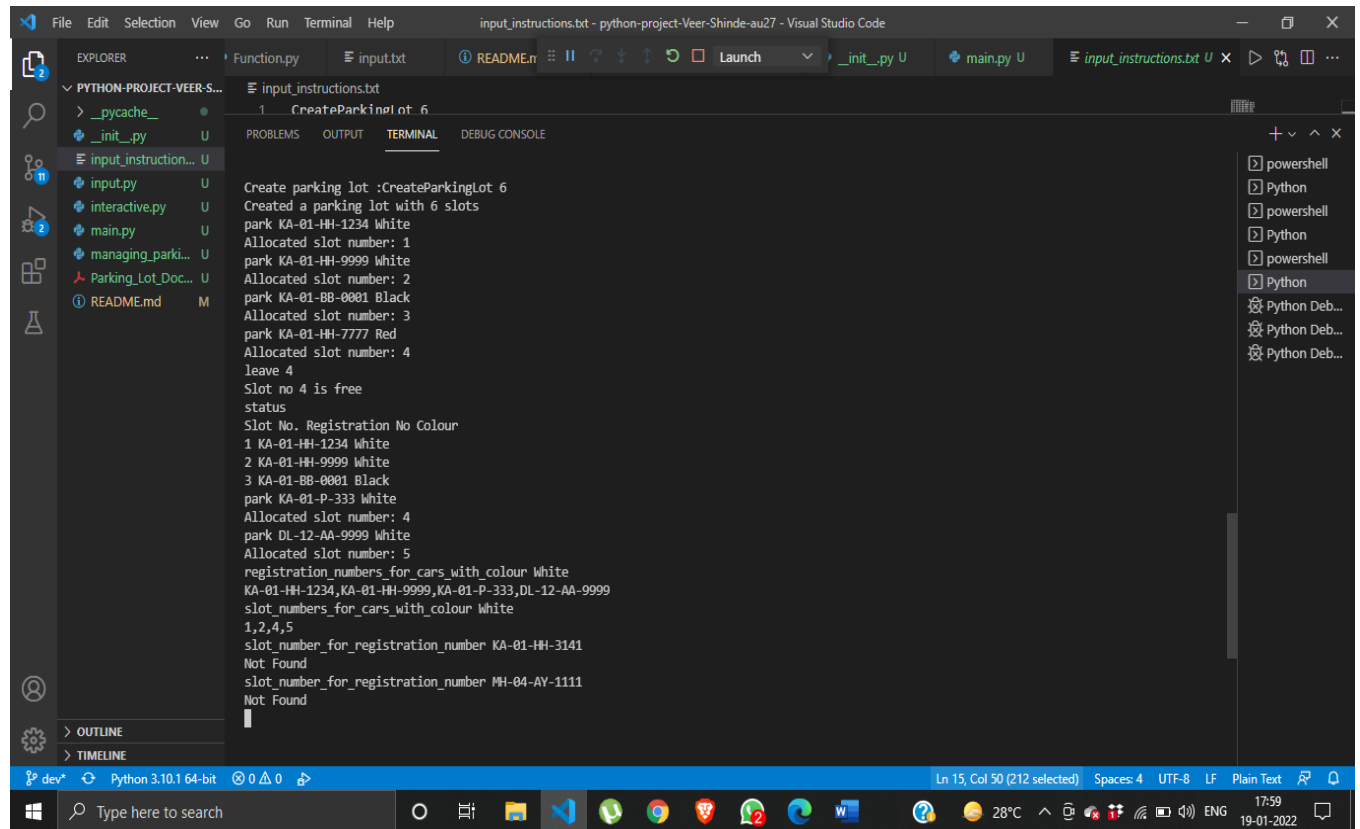
1. **main.py** = Main File (**The file which you have to Run**)
2. **interactive.py** = How my program will run if a user select interactive option.
3. **managing_parking_lot** = The file where I declare and define my class and it's member function.
4. **input.py** = The file where I declare and define non-member function.

Input and Output: **main.py** can be run through shell or through file containing test cases. I did with file handling, in file **input_instructions.txt** all the test cases are there. You just need to run **main.py**, the desired output will be display in your screen. You can change **input_instructions.txt** according to your requirement.

Desired functionality: **main.py** script defines the following functions -

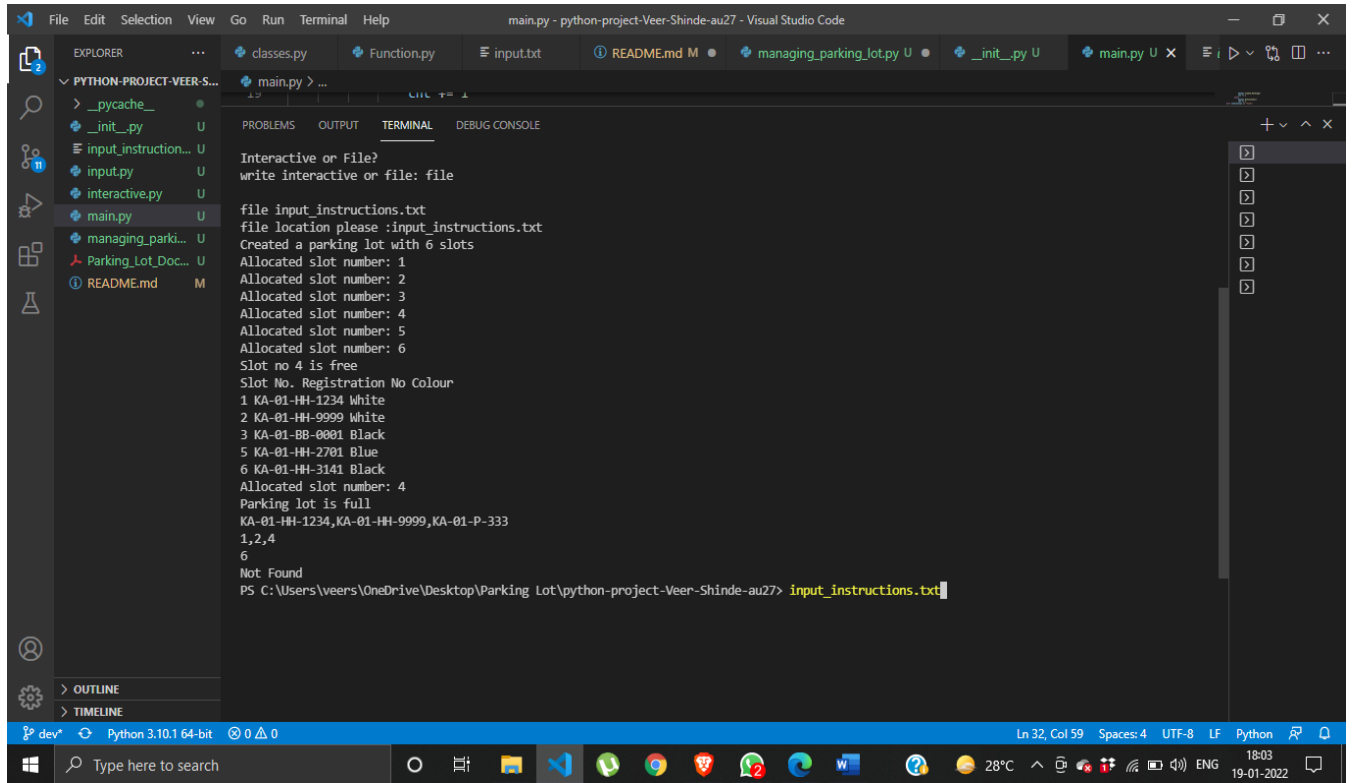
1. **CreateParkingLot n** - Given n number of slots, create a parking lot
2. **park car_regno car_color** - Parks a vehicle with given registration number and color in the nearest empty slot possible. If there are no more empty slots available, it shows a message " Parking lot is full".
3. **status** - Prints the slot number, registration number and color of the parked vehicles.
4. **leave n** - Removes vehicle from slot number n
5. There are few query functions to retrieve slot number from registration number of cars, get registration numbers of cars with particular color etc.

Output Screen for Interactive mode :



```
input_instructions.txt
1 CreateParkingLot 6
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Create parking lot :CreateParkingLot 6
Created a parking lot with 6 slots
park KA-01-HH-1234 White
Allocated slot number: 1
park KA-01-HH-9999 White
Allocated slot number: 2
park KA-01-BB-0001 Black
Allocated slot number: 3
park KA-01-HH-7777 Red
Allocated slot number: 4
leave 4
Slot no 4 is free
status
Slot No. Registration No Colour
1 KA-01-HH-1234 White
2 KA-01-HH-9999 White
3 KA-01-BB-0001 Black
park KA-01-P-333 White
Allocated slot number: 4
park DL-12-AA-9999 White
Allocated slot number: 5
registration numbers for cars with colour White
KA-01-HH-1234,KA-01-HH-9999,KA-01-P-333,DL-12-AA-9999
slot numbers for cars with colour White
1,2,4,5
slot_number_for_registration_number KA-01-HH-3141
Not Found
slot_number_for_registration_number MH-04-AY-1111
Not Found
```

Output Screen for File mode:



The screenshot displays the Visual Studio Code interface with a terminal window open. The terminal shows the output of a Python script executed in file mode. The script prompts for a file location and creates a parking lot with 6 slots. It then allocates slot numbers 1 through 6, with slot 4 being free. The output lists the slot numbers and their corresponding registration numbers and colors. The terminal also shows the command prompt and the file path where the script was executed.

```
main.py - python-project-Veer-Shinde-au27 - Visual Studio Code
EXPLORER
PYTHON-PROJECT-VEER-S...
  > __pycache__
  > __init__.py U
  input_instruction... U
  input.py U
  interactive.py U
  main.py U
  managing_parki... U
  Parking_Lot_Doc... U
  README.md M

TERMINAL
Interactive or File?
write interactive or file: file
file input_instructions.txt
file location please :input_instructions.txt
Created a parking lot with 6 slots
Allocated slot number: 1
Allocated slot number: 2
Allocated slot number: 3
Allocated slot number: 4
Allocated slot number: 5
Allocated slot number: 6
Slot no 4 is free
Slot No. Registration No Colour
1 KA-01-HH-1234 White
2 KA-01-HH-9999 White
3 KA-01-BB-0001 Black
5 KA-01-HH-2701 Blue
6 KA-01-HH-3141 Black
Allocated slot number: 4
Parking lot is full
KA-01-HH-1234,KA-01-HH-9999,KA-01-P-333
1,2,4
6
Not Found
PS C:\Users\veers\OneDrive\Desktop\Parking Lot\python-project-Veer-Shinde-au27> input_instructions.txt
```