Name:Ayush Mishra
Enrollment No. 0801CS211029

Parking Management Project Report

Statistically information about the Parking Management project Program:
Starting Date :11^{th} Nov,2022
Starting Time :10:00 pm
End Date :15^{th} Nov,2022
End time :5:00 pm
Total time required :21 hours
Total line Of Code :369
Total Number of Function:10

Objective of Project:

The objective of this project is to reduces the time in parking of the vehicle i.e. bus , car, auto, etc. and in printing the receipts of charges of vehicle parking.The another objective of program is to Check weather your vehicles is parked in the parking stand or not.And it will help to check insight.

Function description:

1)main function:It is the driver program of the program.From here the program start the execution and call the other function.

2)Park_Veehicle:This function is used to handle parking operation.

3)check_insight:This function use to check parking insight i.e.Total earning and total number of vehicles parked.

4)Bus_parking():This function use to handle bus parking operations.

5)car_park:This function use to handle car parking operations.

6)bike_park():This function use to handle bike parking operations.

7)auto_park():This function use to handle auto parking operations.

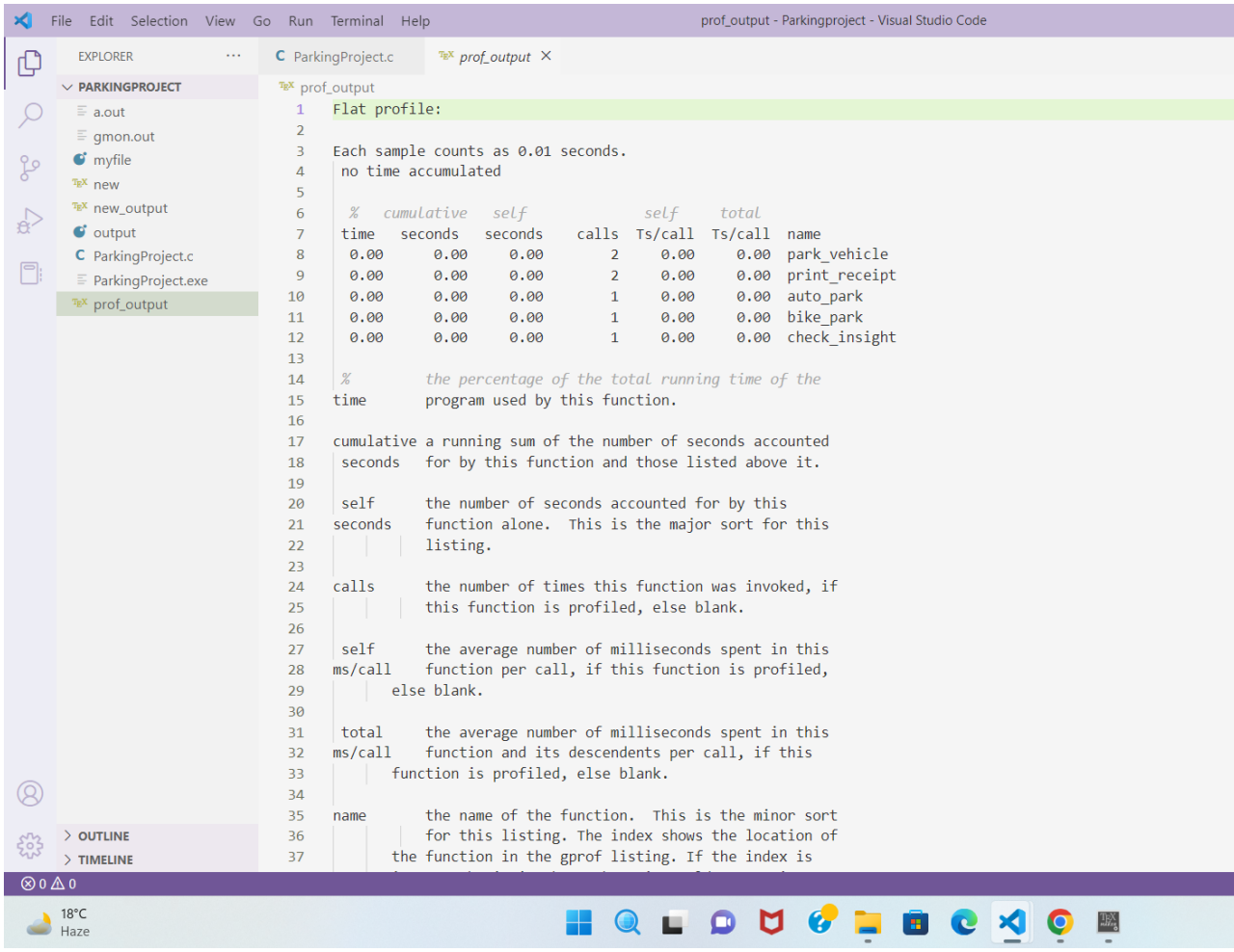
8)cycle_park():This function use to handle cycle parking operations.

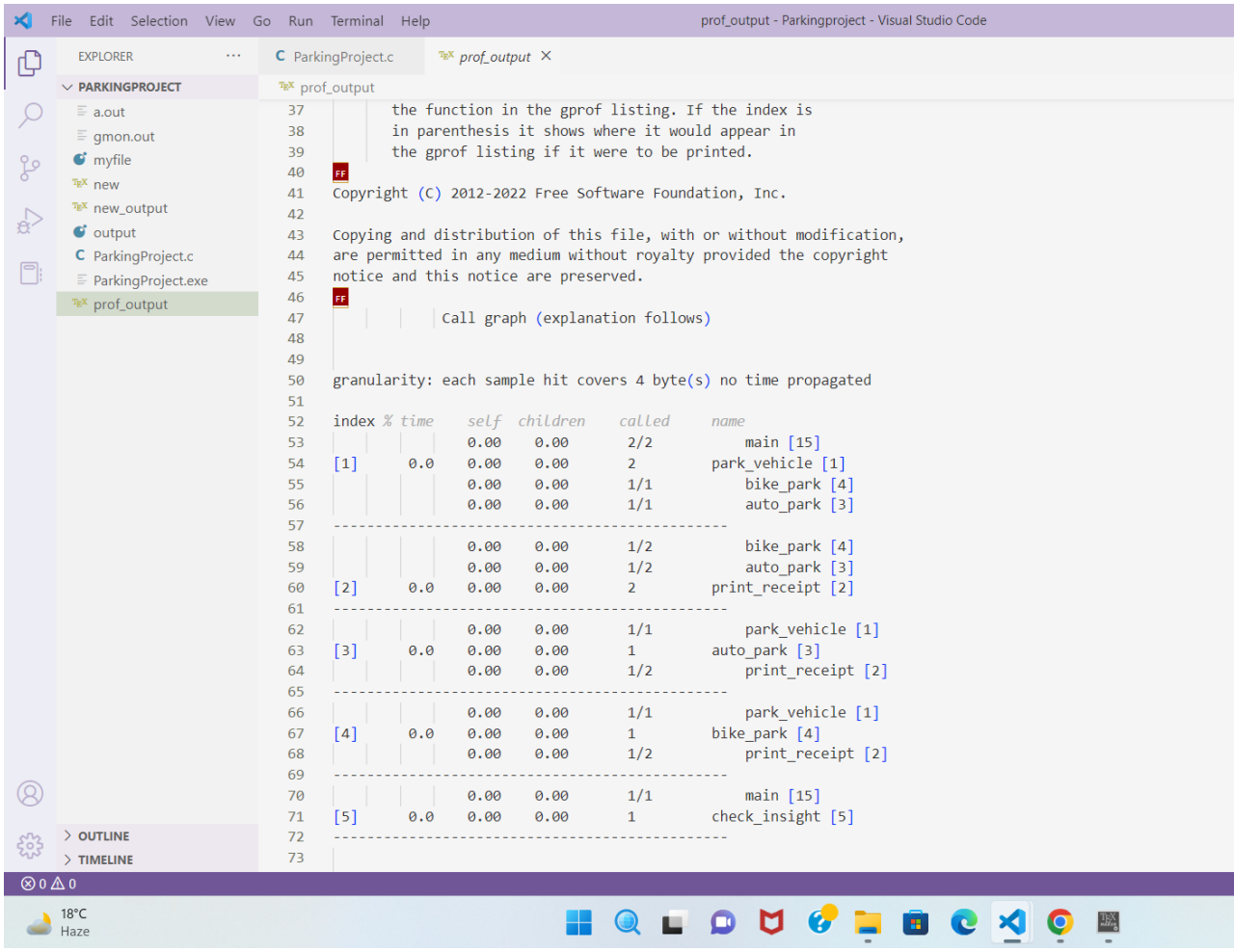
9)truck_park():This function use to handle Truck parking operations.

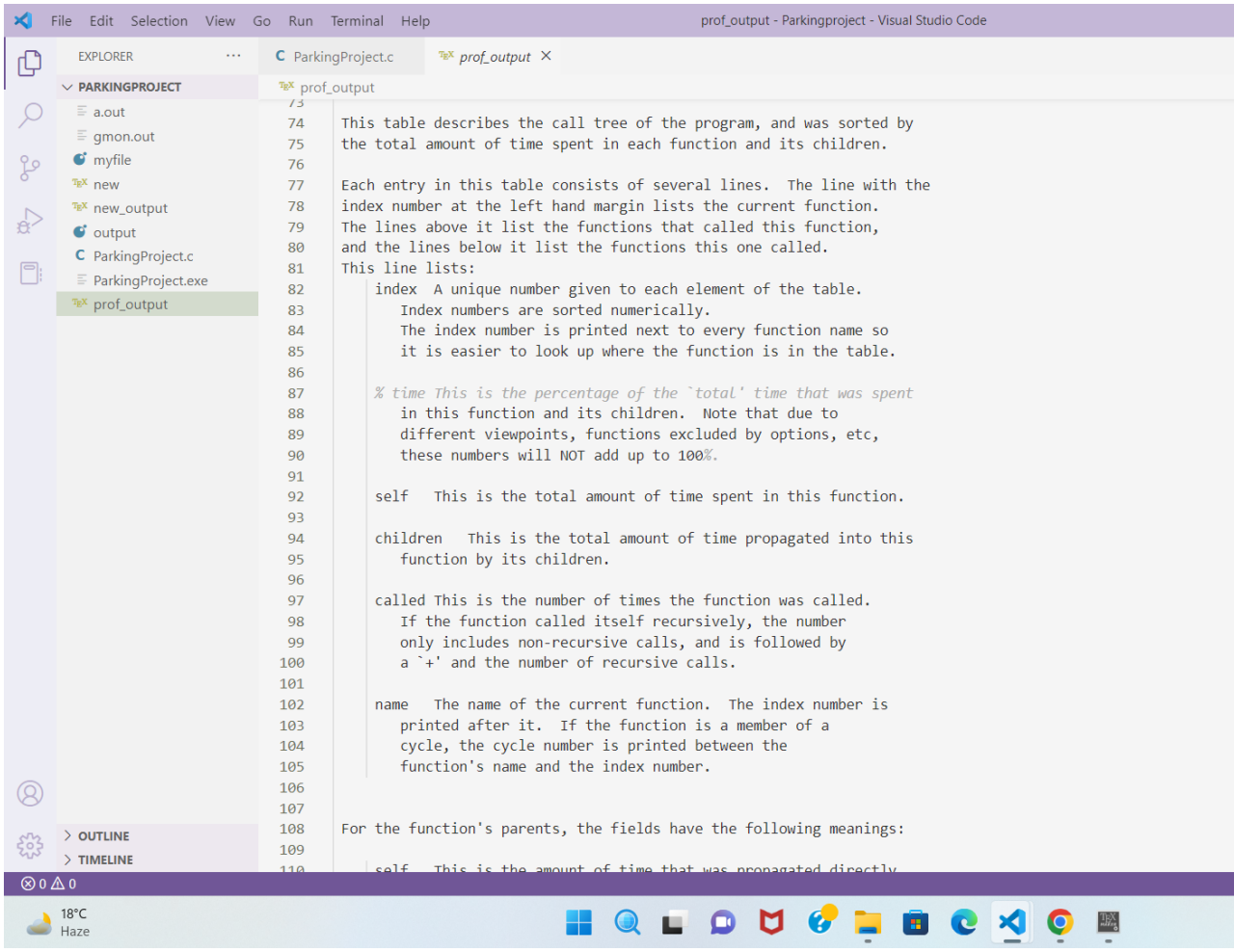
10) print_receipt():This function is to print the receipt.

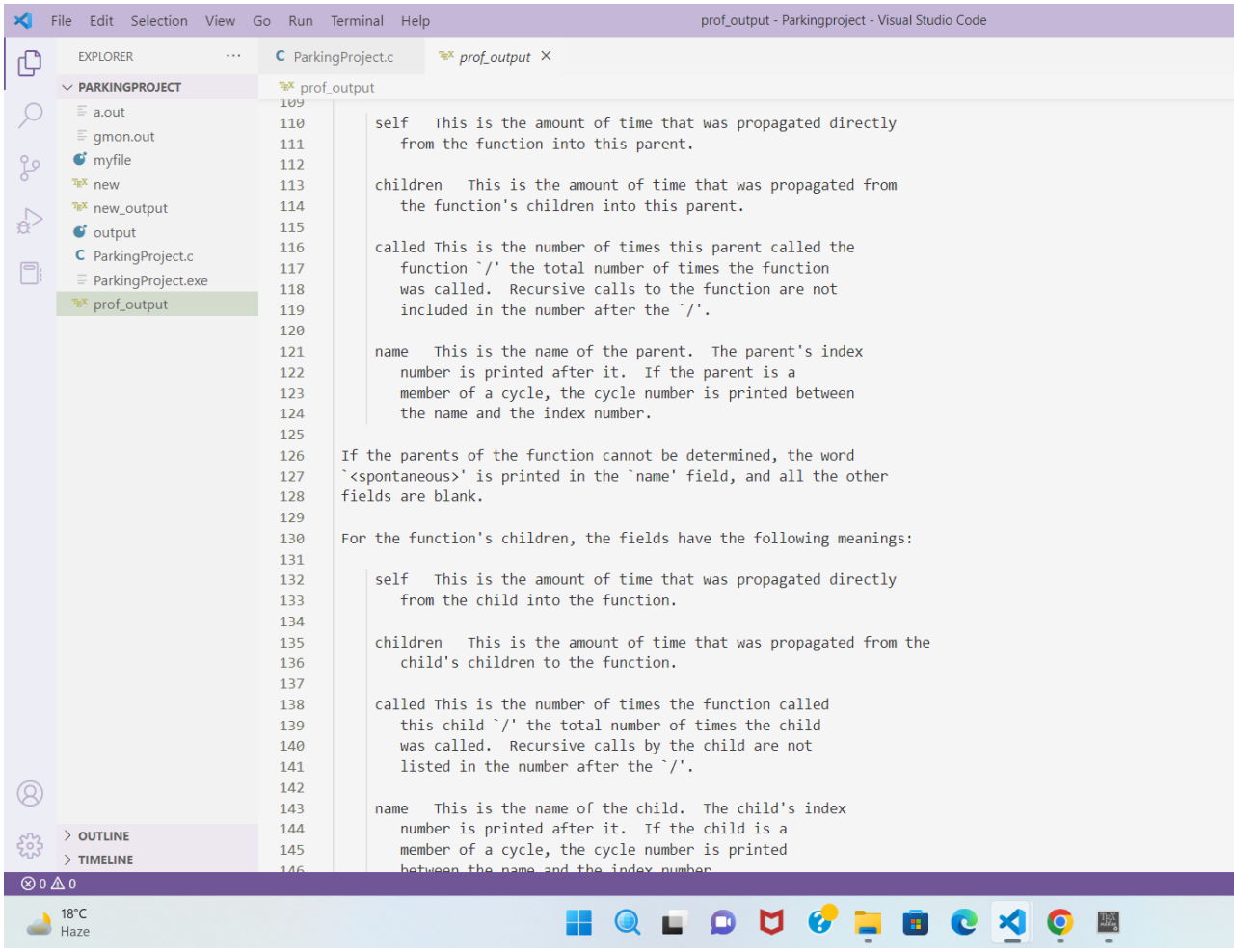
11)exit_function():// this function comment to customer and then exit out.

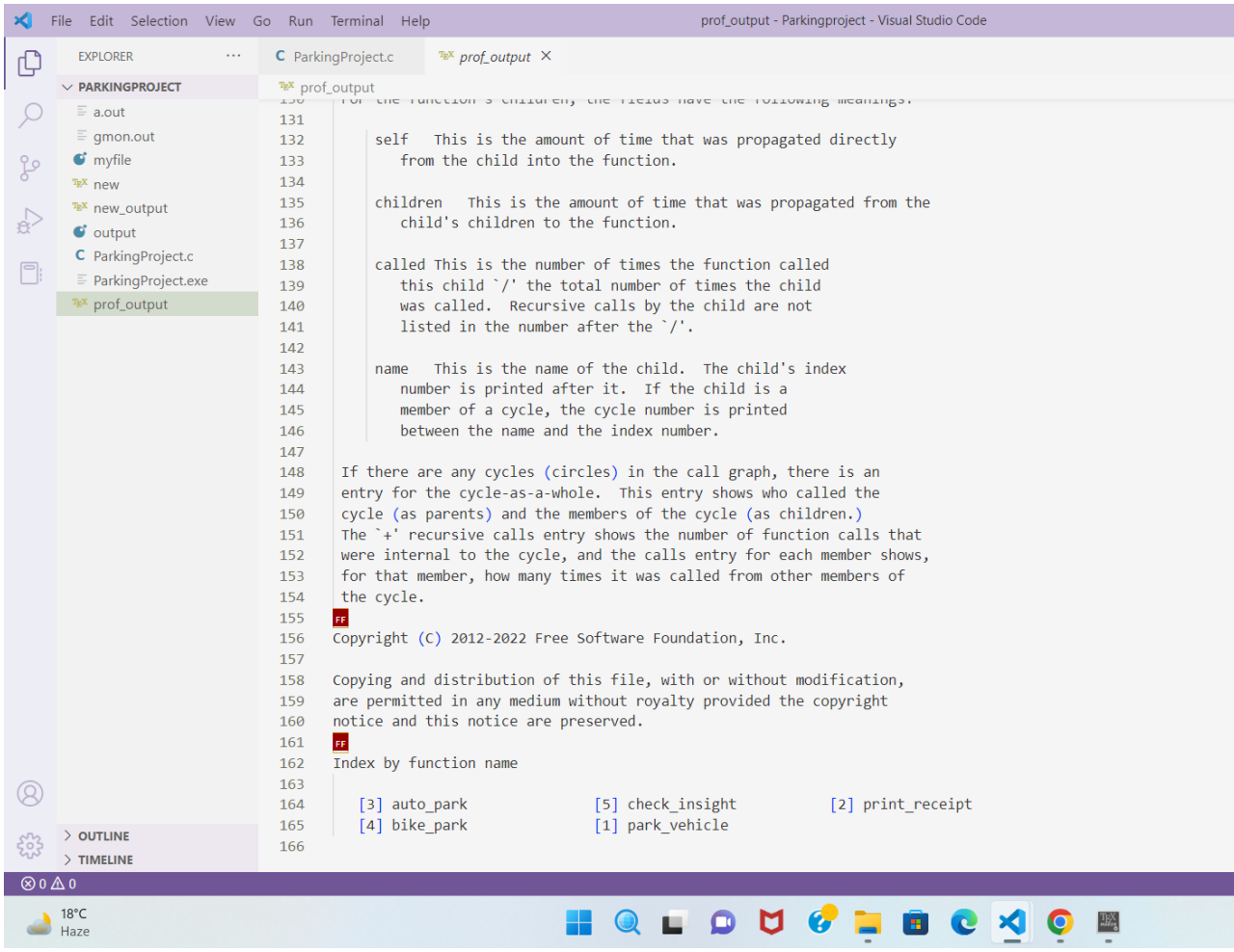
Proffiling


File Edit Selection View Go Run Terminal Help
prof_output - Parkingproject - Visual Studio Code
EXPLORER
PARKINGPROJECT
a.out
gmon.out
myfile
new
new_output
output
ParkingProject.c
ParkingProject.exe
prof_output
OUTLINE
TIMELINE
ParkingProject.c
prof_output
prof_output
Flat profile:
Each sample counts as 0.01 seconds.
no time accumulated
% cumulative self self total
time seconds seconds calls Ts/call Ts/call name
0.00 0.00 0.00 2 0.00 0.00 park_vehicle
0.00 0.00 0.00 2 0.00 0.00 print_receipt
0.00 0.00 0.00 1 0.00 0.00 auto_park
0.00 0.00 0.00 1 0.00 0.00 bike_park
0.00 0.00 0.00 1 0.00 0.00 check_insight
% the percentage of the total running time of the
time program used by this function.
cumulative a running sum of the number of seconds accounted
seconds for by this function and those listed above it.
self the number of seconds accounted for by this
seconds function alone. This is the major sort for this
listing.
calls the number of times this function was invoked, if
this function is profiled, else blank.
self the average number of milliseconds spent in this
ms/call function per call, if this function is profiled,
else blank.
total the average number of milliseconds spent in this
ms/call function and its descendents per call, if this
function is profiled, else blank.
name the name of the function. This is the minor sort
for this listing. The index shows the location of
the function in the gprof listing. If the index is
0 0
18°C
Haze

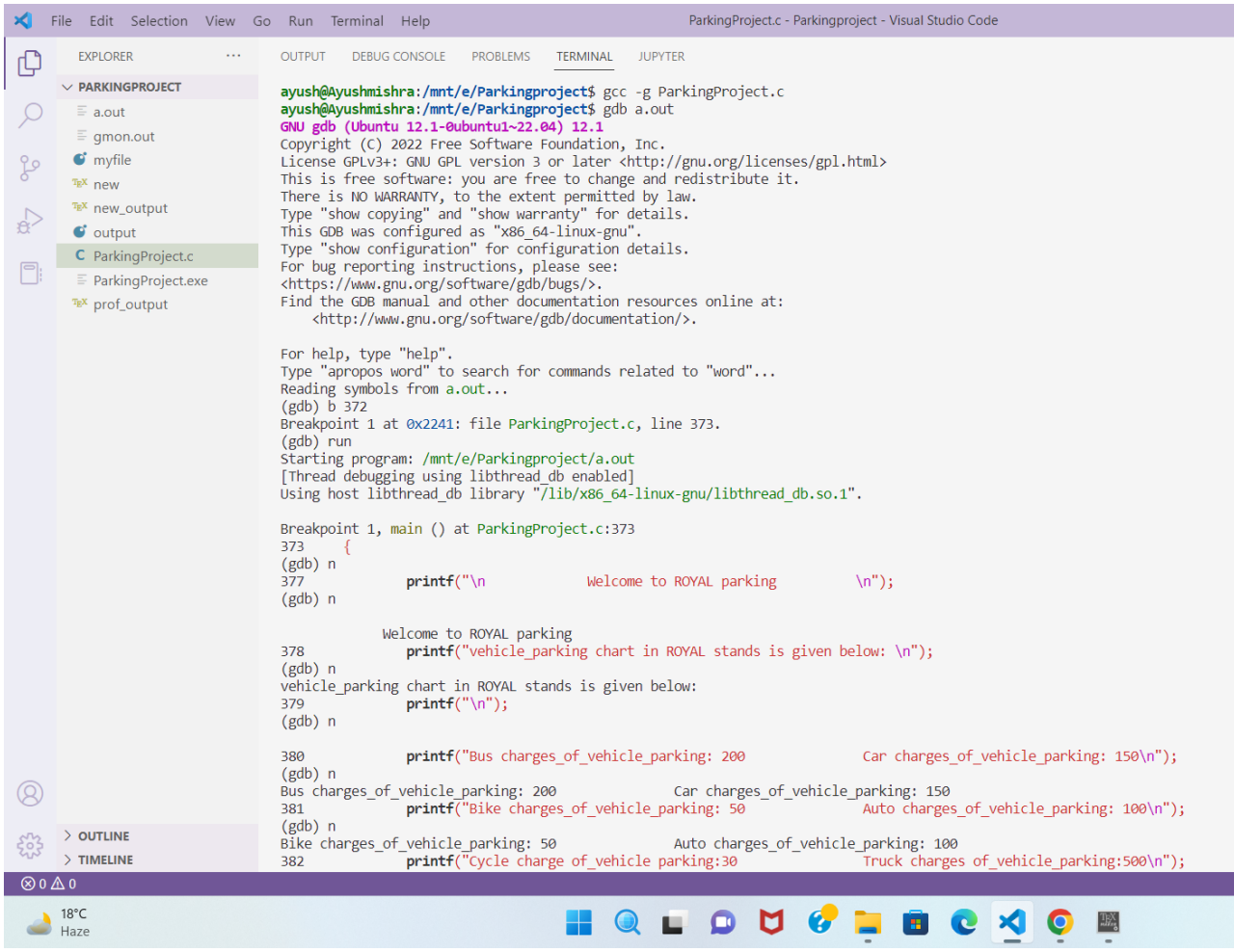

File Edit Selection View Go Run Terminal Help
prof_output - Parkingproject - Visual Studio Code
EXPLORER
PARKINGPROJECT
a.out
gmon.out
myfile
new
new_output
output
ParkingProject.c
ParkingProject.exe
prof_output
OUTLINE
TIMELINE
ParkingProject.c
prof_output
prof_output
the function in the gprof listing. If the index is
in parenthesis it shows where it would appear in
the gprof listing if it were to be printed.
Copyright (C) 2012-2022 Free Software Foundation, Inc.
Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.
Call graph (explanation follows)
granularity: each sample hit covers 4 byte(s) no time propagated
index % time self children called name
0.00 0.00 2/2 main [15]
[1] 0.0 0.00 0.00 2 park_vehicle [1]
0.00 0.00 1/1 bike_park [4]
0.00 0.00 1/1 auto_park [3]
0.00 0.00 1/2 bike_park [4]
0.00 0.00 1/2 auto_park [3]
[2] 0.0 0.00 0.00 2 print_receipt [2]
0.00 0.00 1/1 park_vehicle [1]
[3] 0.0 0.00 0.00 1 auto_park [3]
0.00 0.00 1/2 print_receipt [2]
0.00 0.00 1/1 park_vehicle [1]
[4] 0.0 0.00 0.00 1 bike_park [4]
0.00 0.00 1/2 print_receipt [2]
0.00 0.00 1/1 main [15]
[5] 0.0 0.00 0.00 1 check_insight [5]
0 0
18°C
Haze

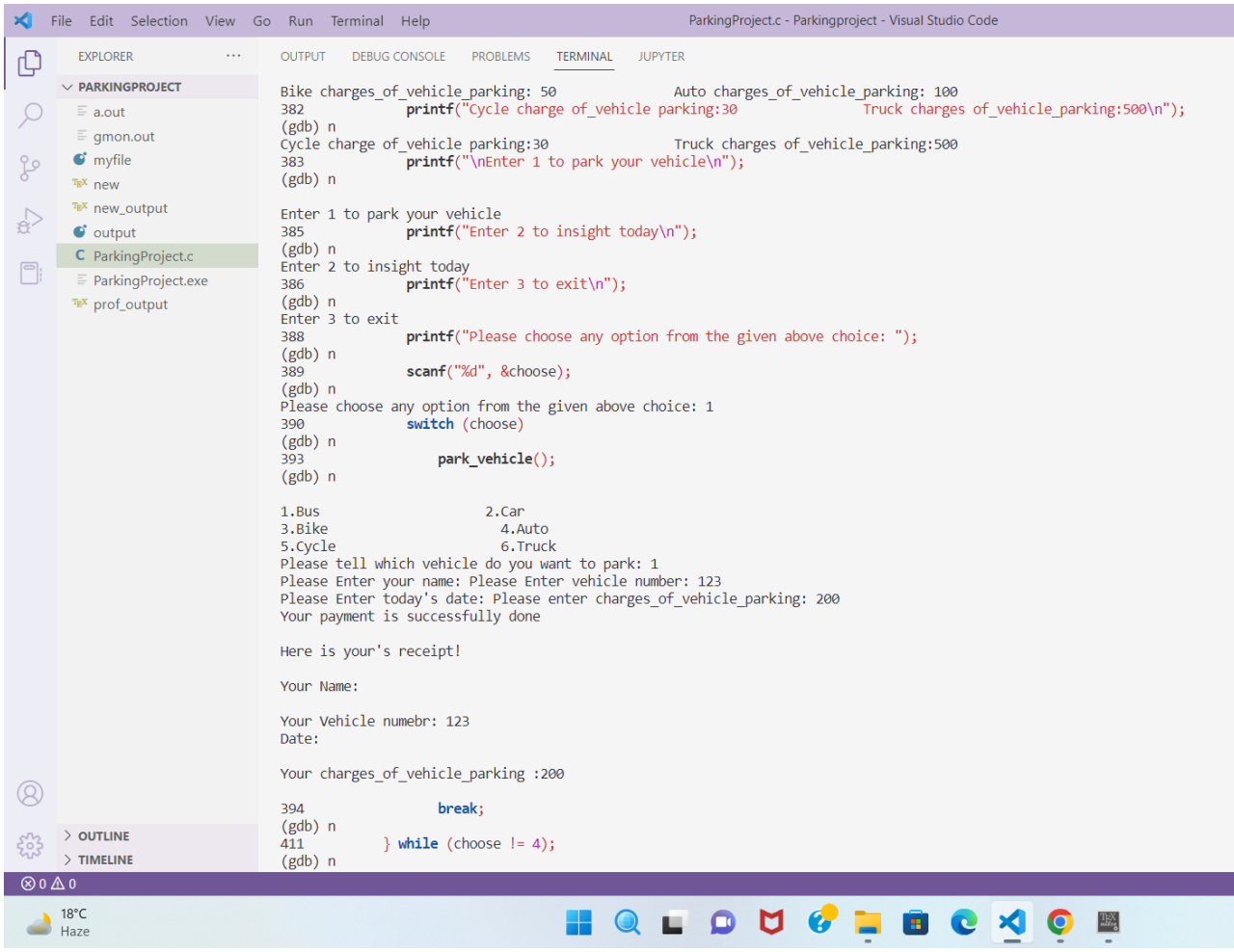

File Edit Selection View Go Run Terminal Help
prof_output - Parkingproject - Visual Studio Code
EXPLORER
PARKINGPROJECT
a.out
gmon.out
myfile
new
new_output
output
ParkingProject.c
ParkingProject.exe
prof_output
OUTLINE
TIMELINE
ParkingProject.c
prof_output
prof_output
This table describes the call tree of the program, and was sorted by
the total amount of time spent in each function and its children.
Each entry in this table consists of several lines. The line with the
index number at the left hand margin lists the current function.
The lines above it list the functions that called this function,
and the lines below it list the functions this one called.
This line lists:
index A unique number given to each element of the table.
Index numbers are sorted numerically.
The index number is printed next to every function name so
it is easier to look up where the function is in the table.
% time This is the percentage of the `total' time that was spent
in this function and its children. Note that due to
different viewpoints, functions excluded by options, etc,
these numbers will NOT add up to 100%.
self This is the total amount of time spent in this function.
children This is the total amount of time propagated into this
function by its children.
called This is the number of times the function was called.
If the function called itself recursively, the number
only includes non-recursive calls, and is followed by
a `+' and the number of recursive calls.
name The name of the current function. The index number is
printed after it. If the function is a member of a
cycle, the cycle number is printed between the
function's name and the index number.
For the function's parents, the fields have the following meanings:
0 0
18°C
Haze

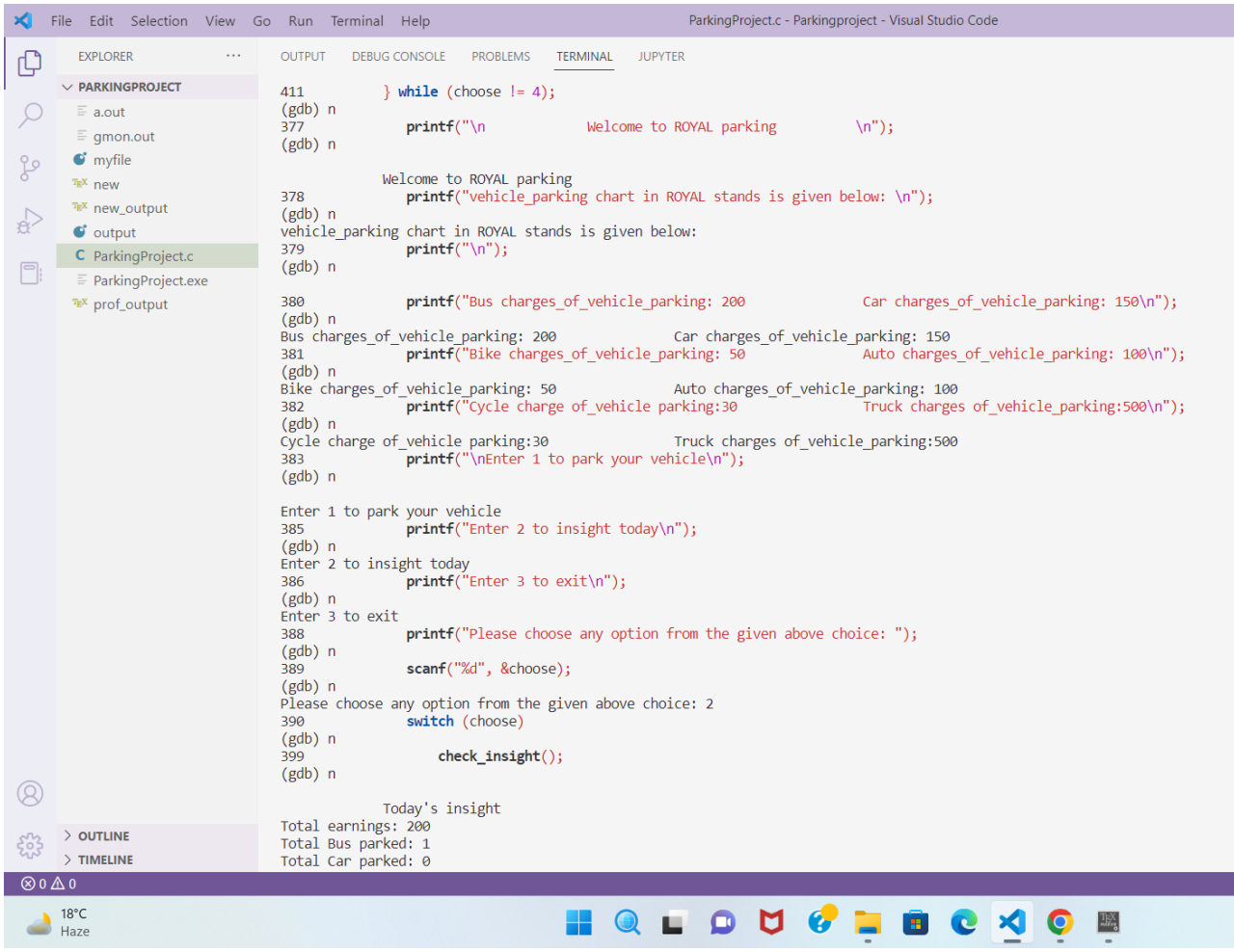

File Edit Selection View Go Run Terminal Help
prof_output - Parkingproject - Visual Studio Code
EXPLORER
PARKINGPROJECT
a.out
gmon.out
myfile
new
new_output
output
ParkingProject.c
ParkingProject.exe
prof_output
OUTLINE
TIMELINE
ParkingProject.c
prof_output
prof_output
self This is the amount of time that was propagated directly
from the function into this parent.
children This is the amount of time that was propagated from
the function's children into this parent.
called This is the number of times this parent called the
function `/' the total number of times the function
was called. Recursive calls to the function are not
included in the number after the `/'.
name This is the name of the parent. The parent's index
number is printed after it. If the parent is a
member of a cycle, the cycle number is printed between
the name and the index number.
If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.
For the function's children, the fields have the following meanings:
self This is the amount of time that was propagated directly
from the child into the function.
children This is the amount of time that was propagated from the
child's children to the function.
called This is the number of times the function called
this child `/' the total number of times the child
was called. Recursive calls by the child are not
listed in the number after the `/'.
name This is the name of the child. The child's index
number is printed after it. If the child is a
member of a cycle, the cycle number is printed
between the name and the index number.
0 0
18°C
Haze

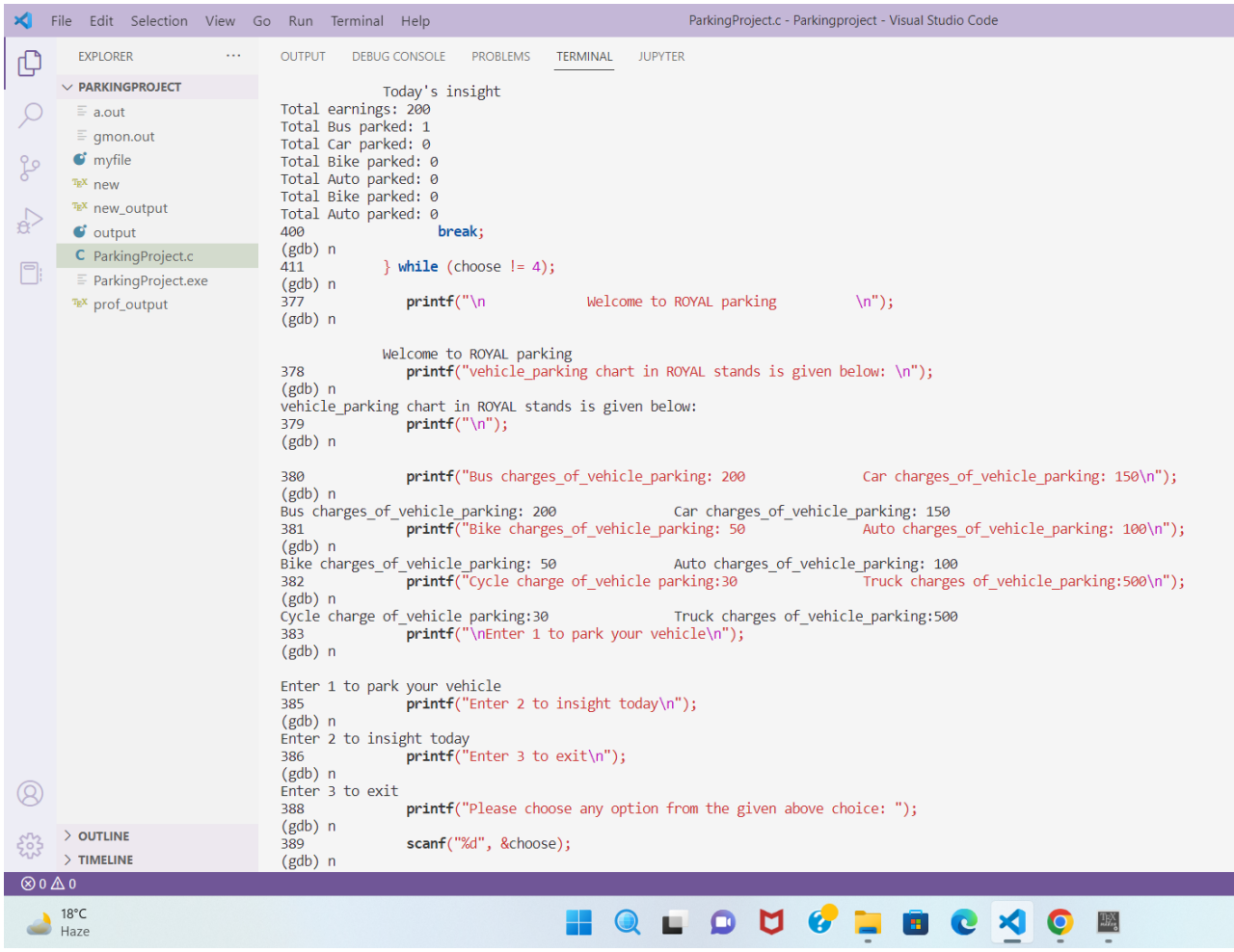

File Edit Selection View Go Run Terminal Help
prof_output - Parkingproject - Visual Studio Code
EXPLORER
PARKINGPROJECT
a.out
gmon.out
myfile
new
new_output
output
ParkingProject.c
ParkingProject.exe
prof_output
OUTLINE
TIMELINE
ParkingProject.c
prof_output
prof_output
self This is the amount of time that was propagated directly
from the child into the function.
children This is the amount of time that was propagated from the
child's children to the function.
called This is the number of times the function called
this child `/' the total number of times the child
was called. Recursive calls by the child are not
listed in the number after the `/'.
name This is the name of the child. The child's index
number is printed after it. If the child is a
member of a cycle, the cycle number is printed
between the name and the index number.
If there are any cycles (circles) in the call graph, there is an
entry for the cycle-as-a-whole. This entry shows who called the
cycle (as parents) and the members of the cycle (as children.)
The `+' recursive calls entry shows the number of function calls that
were internal to the cycle, and the calls entry for each member shows,
for that member, how many times it was called from other members of
the cycle.
Copyright (C) 2012-2022 Free Software Foundation, Inc.
Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.
Index by function name
[3] auto_park
[5] check_insight
[2] print_receipt
[4] bike_park
[1] park_vehicle
0 0
18°C
Haze

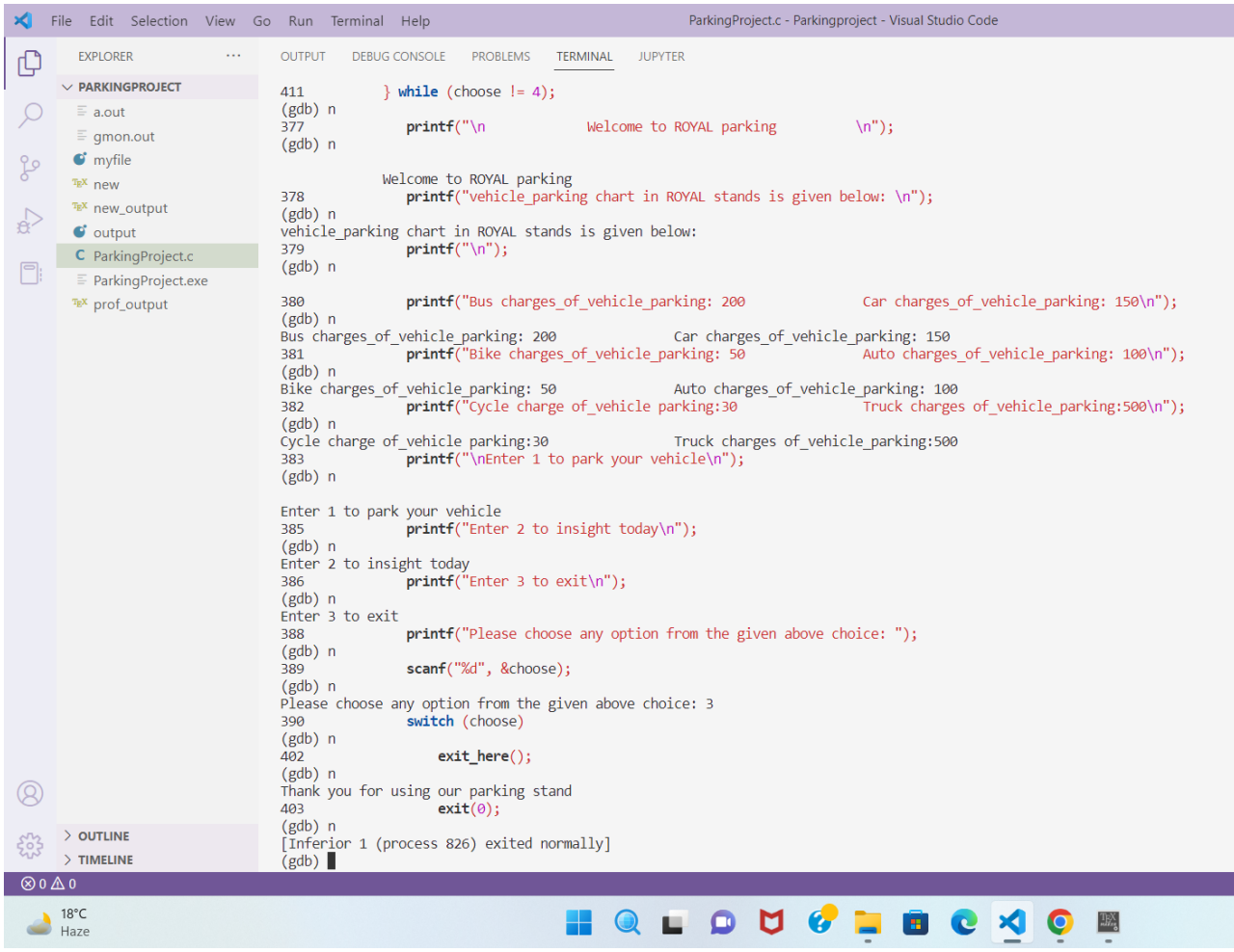
Debugging


File Edit Selection View Go Run Terminal Help
ParkingProject.c - Parkingproject - Visual Studio Code
EXPLORER
PARKINGPROJECT
a.out
gmon.out
myfile
new
new_output
output
ParkingProject.c
ParkingProject.exe
prof_output
OUTLINE
TIMELINE
OUTPUT DEBUG CONSOLE PROBLEMS TERMINAL JUPYTER
ayush@Ayushmishra:/mnt/e/Parkingproject$ gcc -g ParkingProject.c
ayush@Ayushmishra:/mnt/e/Parkingproject$ gdb a.out
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...
(gdb) b 372
Breakpoint 1 at 0x2241: file ParkingProject.c, line 373.
(gdb) run
Starting program: /mnt/e/Parkingproject/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Breakpoint 1, main () at ParkingProject.c:373
373 {
(gdb) n
377 printf("\n Welcome to ROYAL parking \n");
(gdb) n
Welcome to ROYAL parking
378 printf("vehicle_parking chart in ROYAL stands is given below: \n");
(gdb) n
vehicle_parking chart in ROYAL stands is given below:
379 printf("\n");
(gdb) n
380 printf("Bus charges_of_vehicle_parking: 200 Car charges_of_vehicle_parking: 150\n");
(gdb) n
Bus charges_of_vehicle_parking: 200 Car charges_of_vehicle_parking: 150
381 printf("Bike charges_of_vehicle_parking: 50 Auto charges_of_vehicle_parking: 100\n");
(gdb) n
Bike charges_of_vehicle_parking: 50 Auto charges_of_vehicle_parking: 100
382 printf("Cycle charge of_vehicle parking:30 Truck charges of_vehicle_parking:500\n");
0 0
18°C
Haze


File Edit Selection View Go Run Terminal Help
ParkingProject.c - Parkingproject - Visual Studio Code
EXPLORER
PARKINGPROJECT
a.out
gmon.out
myfile
new
new_output
output
ParkingProject.c
ParkingProject.exe
prof_output
OUTLINE
TIMELINE
OUTPUT DEBUG CONSOLE PROBLEMS TERMINAL JUPYTER
Bike charges_of_vehicle_parking: 50 Auto charges_of_vehicle_parking: 100
382 printf("Cycle charge of_vehicle parking:30 Truck charges of_vehicle_parking:500\n");
(gdb) n
Cycle charge of_vehicle parking:30 Truck charges of_vehicle_parking:500
383 printf("\nEnter 1 to park your vehicle\n");
(gdb) n
Enter 1 to park your vehicle
385 printf("Enter 2 to insight today\n");
(gdb) n
Enter 2 to insight today
386 printf("Enter 3 to exit\n");
(gdb) n
Enter 3 to exit
388 printf("Please choose any option from the given above choice: ");
(gdb) n
389 scanf("%d", &choose);
(gdb) n
Please choose any option from the given above choice: 1
390 switch (choose)
(gdb) n
393 park_vehicle();
(gdb) n
1.Bus 2.Car
3.Bike 4.Auto
5.Cycle 6.Truck
Please tell which vehicle do you want to park: 1
Please Enter your name: Please Enter vehicle number: 123
Please Enter today's date: Please enter charges_of_vehicle_parking: 200
Your payment is successfully done
Here is your's receipt!
Your Name:
Your Vehicle numebr: 123
Date:
Your charges_of_vehicle_parking :200
394 break;
(gdb) n
411 } while (choose != 4);
(gdb) n
0 0
18°C
Haze


File Edit Selection View Go Run Terminal Help
ParkingProject.c - Parkingproject - Visual Studio Code
EXPLORER
PARKINGPROJECT
a.out
gmon.out
myfile
new
new_output
output
ParkingProject.c
ParkingProject.exe
prof_output
OUTLINE
TIMELINE
OUTPUT DEBUG CONSOLE PROBLEMS TERMINAL JUPYTER
411 } while (choose != 4);
(gdb) n
377 printf("\n Welcome to ROYAL parking \n");
(gdb) n
Welcome to ROYAL parking
378 printf("vehicle_parking chart in ROYAL stands is given below: \n");
(gdb) n
vehicle_parking chart in ROYAL stands is given below:
379 printf("\n");
(gdb) n
380 printf("Bus charges_of_vehicle_parking: 200 Car charges_of_vehicle_parking: 150\n");
(gdb) n
Bus charges_of_vehicle_parking: 200 Car charges_of_vehicle_parking: 150
381 printf("Bike charges_of_vehicle_parking: 50 Auto charges_of_vehicle_parking: 100\n");
(gdb) n
Bike charges_of_vehicle_parking: 50 Auto charges_of_vehicle_parking: 100
382 printf("Cycle charge of_vehicle parking:30 Truck charges of_vehicle_parking:500\n");
(gdb) n
Cycle charge of_vehicle parking:30 Truck charges of_vehicle_parking:500
383 printf("\nEnter 1 to park your vehicle\n");
(gdb) n
Enter 1 to park your vehicle
385 printf("Enter 2 to insight today\n");
(gdb) n
Enter 2 to insight today
386 printf("Enter 3 to exit\n");
(gdb) n
Enter 3 to exit
388 printf("Please choose any option from the given above choice: ");
(gdb) n
389 scanf("%d", &choose);
(gdb) n
Please choose any option from the given above choice: 2
390 switch (choose)
(gdb) n
399 check_insight();
(gdb) n
Today's insight
Total earnings: 200
Total Bus parked: 1
Total Car parked: 0
0 0
18°C
Haze


File Edit Selection View Go Run Terminal Help
ParkingProject.c - Parkingproject - Visual Studio Code
EXPLORER
PARKINGPROJECT
a.out
gmon.out
myfile
new
new_output
output
ParkingProject.c
ParkingProject.exe
prof_output
OUTLINE
TIMELINE
OUTPUT DEBUG CONSOLE PROBLEMS TERMINAL JUPYTER
Today's insight
Total earnings: 200
Total Bus parked: 1
Total Car parked: 0
Total Bike parked: 0
Total Auto parked: 0
Total Bike parked: 0
Total Auto parked: 0
400 break;
(gdb) n
411 } while (choose != 4);
(gdb) n
377 printf("\n Welcome to ROYAL parking \n");
(gdb) n
Welcome to ROYAL parking
378 printf("vehicle_parking chart in ROYAL stands is given below: \n");
(gdb) n
vehicle_parking chart in ROYAL stands is given below:
379 printf("\n");
(gdb) n
380 printf("Bus charges_of_vehicle_parking: 200 Car charges_of_vehicle_parking: 150\n");
(gdb) n
Bus charges_of_vehicle_parking: 200 Car charges_of_vehicle_parking: 150
381 printf("Bike charges_of_vehicle_parking: 50 Auto charges_of_vehicle_parking: 100\n");
(gdb) n
Bike charges_of_vehicle_parking: 50 Auto charges_of_vehicle_parking: 100
382 printf("Cycle charge of_vehicle parking:30 Truck charges of_vehicle_parking:500\n");
(gdb) n
Cycle charge of_vehicle parking:30 Truck charges of_vehicle_parking:500
383 printf("\nEnter 1 to park your vehicle\n");
(gdb) n
Enter 1 to park your vehicle
385 printf("Enter 2 to insight today\n");
(gdb) n
Enter 2 to insight today
386 printf("Enter 3 to exit\n");
(gdb) n
Enter 3 to exit
388 printf("Please choose any option from the given above choice: ");
(gdb) n
389 scanf("%d", &choose);
(gdb) n
0 0
18°C
Haze


File Edit Selection View Go Run Terminal Help
ParkingProject.c - Parkingproject - Visual Studio Code
EXPLORER
PARKINGPROJECT
a.out
gmon.out
myfile
new
new_output
output
ParkingProject.c
ParkingProject.exe
prof_output
OUTLINE
TIMELINE
OUTPUT DEBUG CONSOLE PROBLEMS TERMINAL JUPYTER
411 } while (choose != 4);
(gdb) n
377 printf("\n Welcome to ROYAL parking \n");
(gdb) n
Welcome to ROYAL parking
378 printf("vehicle_parking chart in ROYAL stands is given below: \n");
(gdb) n
vehicle_parking chart in ROYAL stands is given below:
379 printf("\n");
(gdb) n
380 printf("Bus charges_of_vehicle_parking: 200 Car charges_of_vehicle_parking: 150\n");
(gdb) n
Bus charges_of_vehicle_parking: 200 Car charges_of_vehicle_parking: 150
381 printf("Bike charges_of_vehicle_parking: 50 Auto charges_of_vehicle_parking: 100\n");
(gdb) n
Bike charges_of_vehicle_parking: 50 Auto charges_of_vehicle_parking: 100
382 printf("Cycle charge of_vehicle parking:30 Truck charges of_vehicle_parking:500\n");
(gdb) n
Cycle charge of_vehicle parking:30 Truck charges of_vehicle_parking:500
383 printf("\nEnter 1 to park your vehicle\n");
(gdb) n
Enter 1 to park your vehicle
385 printf("Enter 2 to insight today\n");
(gdb) n
Enter 2 to insight today
386 printf("Enter 3 to exit\n");
(gdb) n
Enter 3 to exit
388 printf("Please choose any option from the given above choice: ");
(gdb) n
389 scanf("%d", &choose);
(gdb) n
Please choose any option from the given above choice: 3
390 switch (choose)
(gdb) n
402 exit_here();
(gdb) n
Thank you for using our parking stand
403 exit(0);
(gdb) n
[Inferior 1 (process 826) exited normally]
(gdb)
0 0
18°C
Haze

C project Code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int Bus_Places = 0, Car_Places = 5, Bike_Places = 20, Auto_places = 15, Cycle_pl

int earnings = 0;

int BusCount = 0, CarCount = 0, BikeCount = 0, AutoCount = 0, CycleCount = 0, Tr

int index1 = 0, index2 = 0, index3 = 0; // index1 is for vehicle number & index2

struct customer
{
    char customerName[10][20], date[10][13];
    int vehicleNumber[10];
} data; // variable data is declared with 'Point'.

// this function is to print the receipt
void print_receipt(int charges_of_vehicle_parking)
{
    printf("\nHere is your's receipt!\n");
    printf("#########################\n");
    printf("Your Name: %s", data.customerName[index2]);
    printf("Your Vehicle numebr: %d\n", data.vehicleNumber[index1]);
    printf("Date: %s\n", data.date[index3]);
    printf("Your charges_of_vehicle_parking :%d\n", charges_of_vehicle_parking);
    printf("#########################\n");
}

// this function use to handle bus parking operations
void Bus_parking()
{
    int charges_of_vehicle_parking;
    printf("Please Enter your name: ");
    fflush(stdin); // fflush() is used to immediately flush out the contents of
    fgets(data.customerName[index2],20,stdin);
    printf("Please Enter vehicle number: ");
    scanf("%d", &data.vehicleNumber[index1]);
    printf("Please Enter today's date: ");
    fflush(stdin);
    fgets(data.date[index3],13,stdin);
repeat:
    printf("Please enter charges_of_vehicle_parking: ");
```

```
    scanf("%d", &charges_of_vehicle_parking);
    if (charges_of_vehicle_parking < 200 || charges_of_vehicle_parking > 200)
    {
        printf("Please enter valid amount");
        goto repeat;

    }
    else
    {
        earnings = earnings + charges_of_vehicle_parking;
        printf("Your payment is successfully done\n");
    }
    print_receipt(charges_of_vehicle_parking);
    index1++;
    index2++;
    index3++;
    BusCount++;
}

// this function use to handle car parking operations
void car_park()
{
    int charges_of_vehicle_parking;
    printf("Please Enter your name: ");
    fflush(stdin);
    fgets(data.customerName[index2],20,stdin);
    printf("Please Enter vehicle number: ");
    scanf("%d", &data.vehicleNumber[index1]);
    printf("Please Enter today's date: ");
    fflush(stdin);
    fgets(data.date[index3],13,stdin);
repeat:
    printf("Enter parking charges_of_vehicle_parking: ");
    scanf("%d", &charges_of_vehicle_parking);
    if (charges_of_vehicle_parking < 150 || charges_of_vehicle_parking > 150)
    {
        printf("Please enter valid amount");
       goto repeat;

    }
    else
    {
        earnings = earnings + charges_of_vehicle_parking;
        printf("Your payment is successfully done\n");
    }
    print_receipt(charges_of_vehicle_parking);
```

```
    index1++;
    index2++;
    index3++;
    CarCount++;
}

// this function use to handle bike parking operations
void bike_park()
{
    int charges_of_vehicle_parking;
    printf("Please Enter your name: ");
    fflush(stdin);
    fgets(data.customerName[index2],20,stdin);
    printf("Please Enter vehicle number: ");
    scanf("%d", &data.vehicleNumber[index1]);
    printf("please Enter today's date: ");
    fflush(stdin);
    fgets(data.date[index3],13,stdin);
repeat:
    printf("Enter parking charges_of_vehicle_parking: ");
    scanf("%d", &charges_of_vehicle_parking);
    if (charges_of_vehicle_parking < 50 || charges_of_vehicle_parking > 50)
    {
        printf("Please enter valid amount");
        goto repeat;

    }
    else
    {
        earnings = earnings + charges_of_vehicle_parking;
        printf("Your payment is successfully done\n");
    }
    print_receipt(charges_of_vehicle_parking);
    index1++;
    index2++;
    index3++;
    BikeCount++;
}

// this function use to handle auto parking operations
void auto_park()
{
    int charges_of_vehicle_parking;
    printf("Please Enter your name: ");
    fflush(stdin);
    fgets(data.customerName[index2],20,stdin);
```

```
    printf("Please Enter vehicle number: ");
    scanf("%d", &data.vehicleNumber[index1]);
    printf("Please Enter today's date: ");
    fflush(stdin);
    fgets(data.date[index3],13,stdin);
repeat:
    printf("Enter parking charges_of_vehicle_parking: ");
    scanf("%d", &charges_of_vehicle_parking);
    if (charges_of_vehicle_parking < 100 || charges_of_vehicle_parking > 100)
    {
        printf("Please enter valid amount");
        goto repeat;

    }
    else
    {
        earnings = earnings + charges_of_vehicle_parking;
        printf("Your payment is successfully done\n");
    }
    print_receipt(charges_of_vehicle_parking);
    index1++;
    index2++;
    index3++;
    AutoCount++;
}

// this function use to handle cycle parking operations
void cycle_park()
{
    int charges_of_vehicle_parking;
    printf("Please Enter your name: ");
    fflush(stdin);
    fgets(data.customerName[index2],20,stdin);
    printf("Please Enter vehicle number: ");
    scanf("%d", &data.vehicleNumber[index1]);
    printf("Please Enter today's date: ");
    fflush(stdin);
    fgets(data.date[index3],13,stdin);
repeat:
    printf("Enter parking charges_of_vehicle_parking: ");
    scanf("%d", &charges_of_vehicle_parking);
    if (charges_of_vehicle_parking < 30 || charges_of_vehicle_parking > 30)
    {
        printf("Please enter valid amount");
        goto repeat;
```

```
    }
    else
    {
        earnings = earnings + charges_of_vehicle_parking;
        printf("Your payment is successfully done\n");
    }
    print_receipt(charges_of_vehicle_parking);
    index1++;
    index2++;
    index3++;
    CycleCount++;
}

// this function use to handle Truck parking operations
void truck_park()
{
    int charges_of_vehicle_parking;
    printf("Please Enter your name: ");
    fflush(stdin);
    fgets(data.customerName[index2],20,stdin);
    printf("Please Enter vehicle number: ");
    scanf("%d", &data.vehicleNumber[index1]);
    printf("Please Enter today's date: ");
    fflush(stdin);
    fgets(data.date[index3],13,stdin);
repeat:
    printf("Enter parking charges_of_vehicle_parking: ");
    scanf("%d", &charges_of_vehicle_parking);
    if (charges_of_vehicle_parking < 500 || charges_of_vehicle_parking > 500)
    {
        printf("Please enter valid amount");
        goto repeat;

    }
    else
    {
        earnings = earnings + charges_of_vehicle_parking;
        printf("Your payment is successfully done\n");
    }
    print_receipt(charges_of_vehicle_parking);
    index1++;
    index2++;
    index3++;
    TruckCount++;
}
```

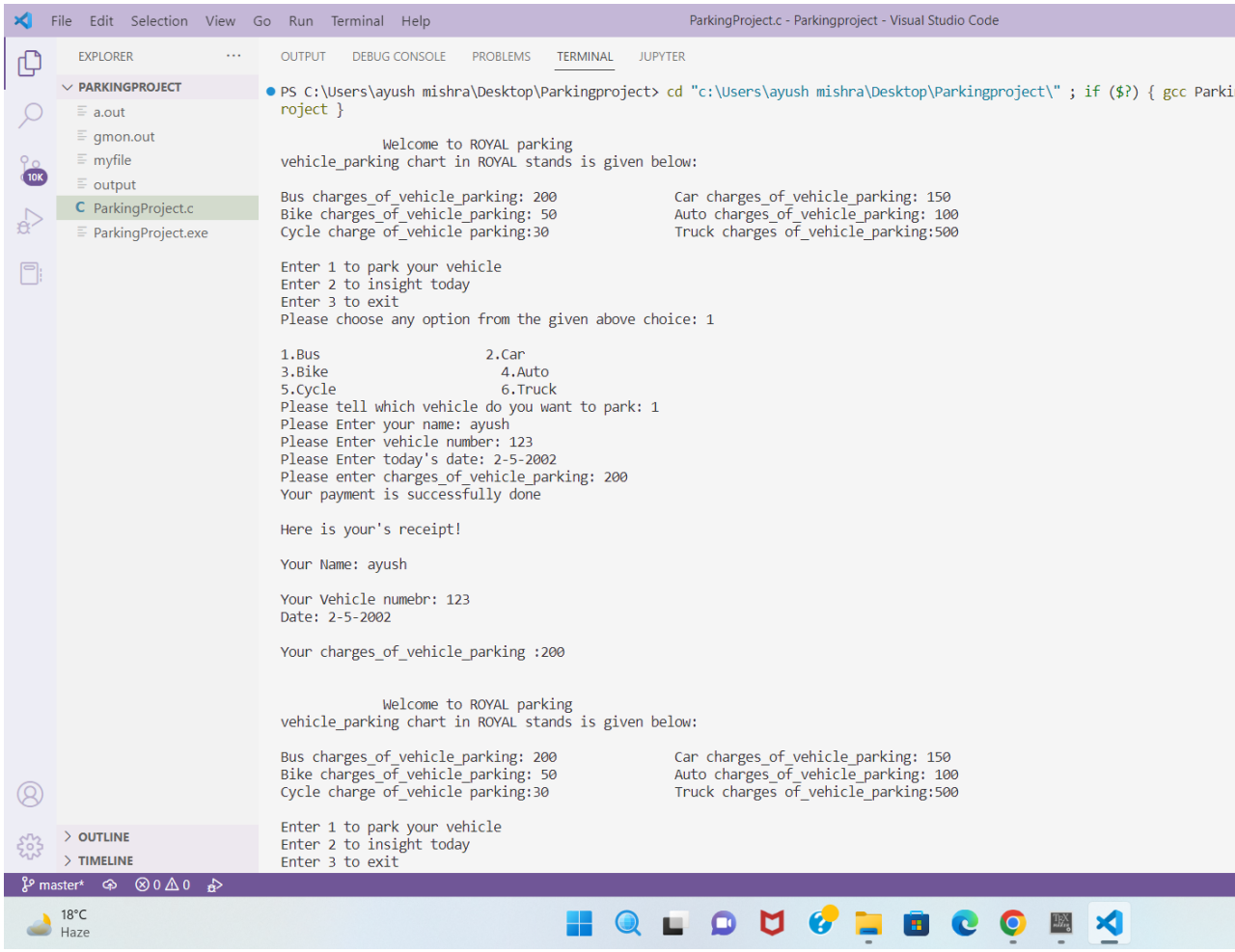
```
// This function is used to handle parking operation
void park_vehicle()
{
    int choose;
again:
    printf("\n1.Bus                    2.Car\n");
    printf("3.Bike                   4.Auto\n");
    printf("5.Cycle                  6.Truck\n");
    printf("Please tell which vehicle do you want to park: ");
    scanf("%d", &choose);
    switch (choose)
    {
    case 1:
        if (Bus_Places == 0)
        {
            printf("#########################");
            printf("\nSorry! Bus parking slot is not available\n");
            printf("#########################\n");
        }
        else
        {
            Bus_parking();
            Bus_Places--;
        }
        break;
    case 2:
        if (Car_Places == 0)
        {
            printf("#########################\n");
            printf("\nSorry! Car parking slot is not available\n");
            printf("#########################\n");
        }
        else
        {
            car_park();
            Car_Places--;
        }
        break;
    case 3:
        if (Bike_Places == 0)
        {
            printf("#########################\n");
            printf("\nSorry! Bike parking slot is not available\n");
            printf("#########################\n");
        }
        else
```

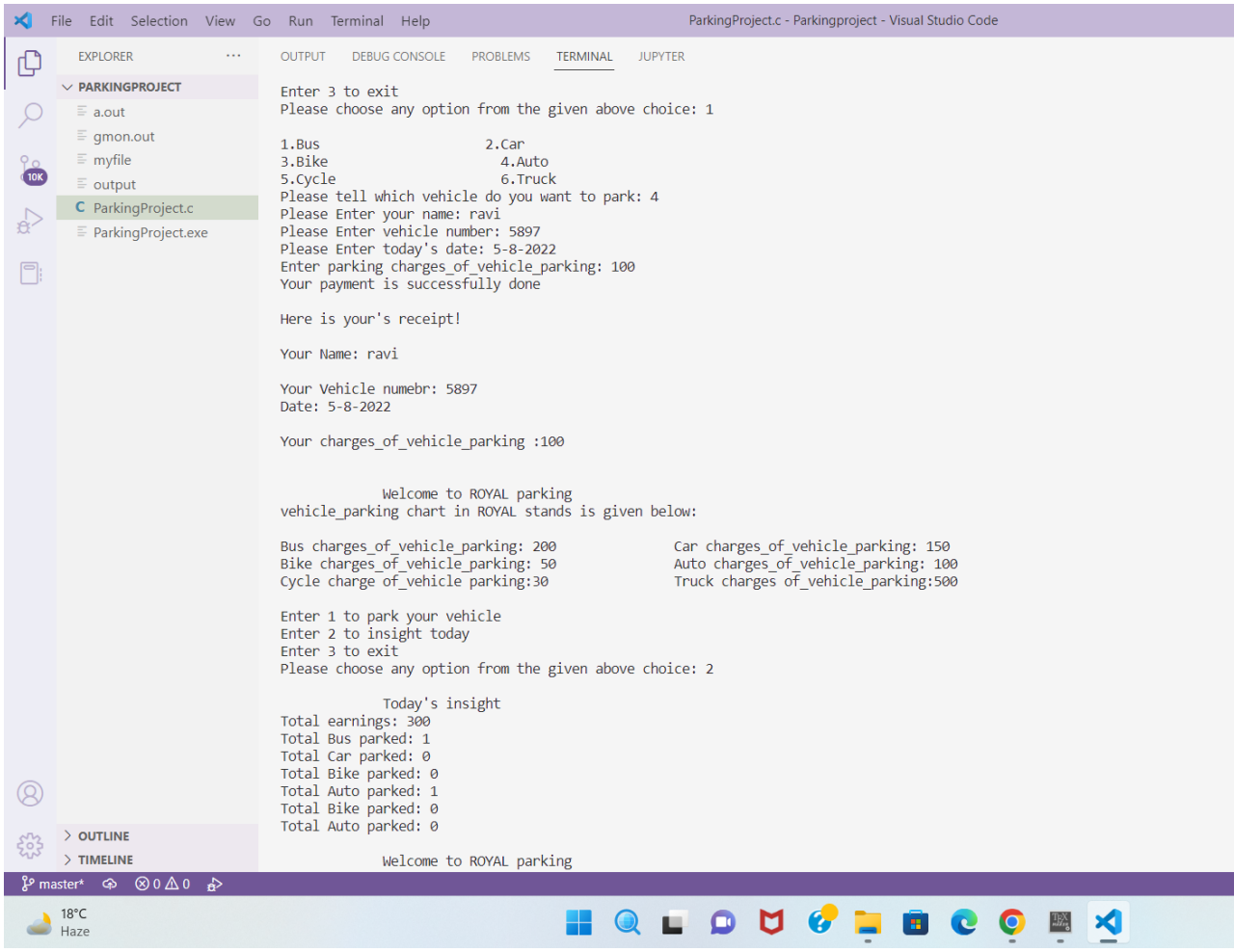
```
    {
        bike_park();
        Bike_Places--;
    }
    break;
case 4:
    if (Auto_places == 0)
    {
        printf("##########################\n");
        printf("\nSorry! Auto parking slot is not available\n");
        printf("##########################\n");
    }
    else
    {
        auto_park();
        Auto_places--;
    }
    break;
case 5:
    if (Cycle_places == 0)
    {
        printf("##########################\n");
        printf("\nSorry! Auto parking slot is not available\n");
        printf("##########################\n");
    }
    else
    {
        cycle_park();
        Cycle_places--;
    }
    break;
case 6:
    if (Truck_places == 0)
    {
        printf("##########################\n");
        printf("\nSorry! Auto parking slot is not available\n");
         printf("##########################\n");

    }
    else
    {
        truck_park();
        Truck_places--;
    }
    break;
default:
```

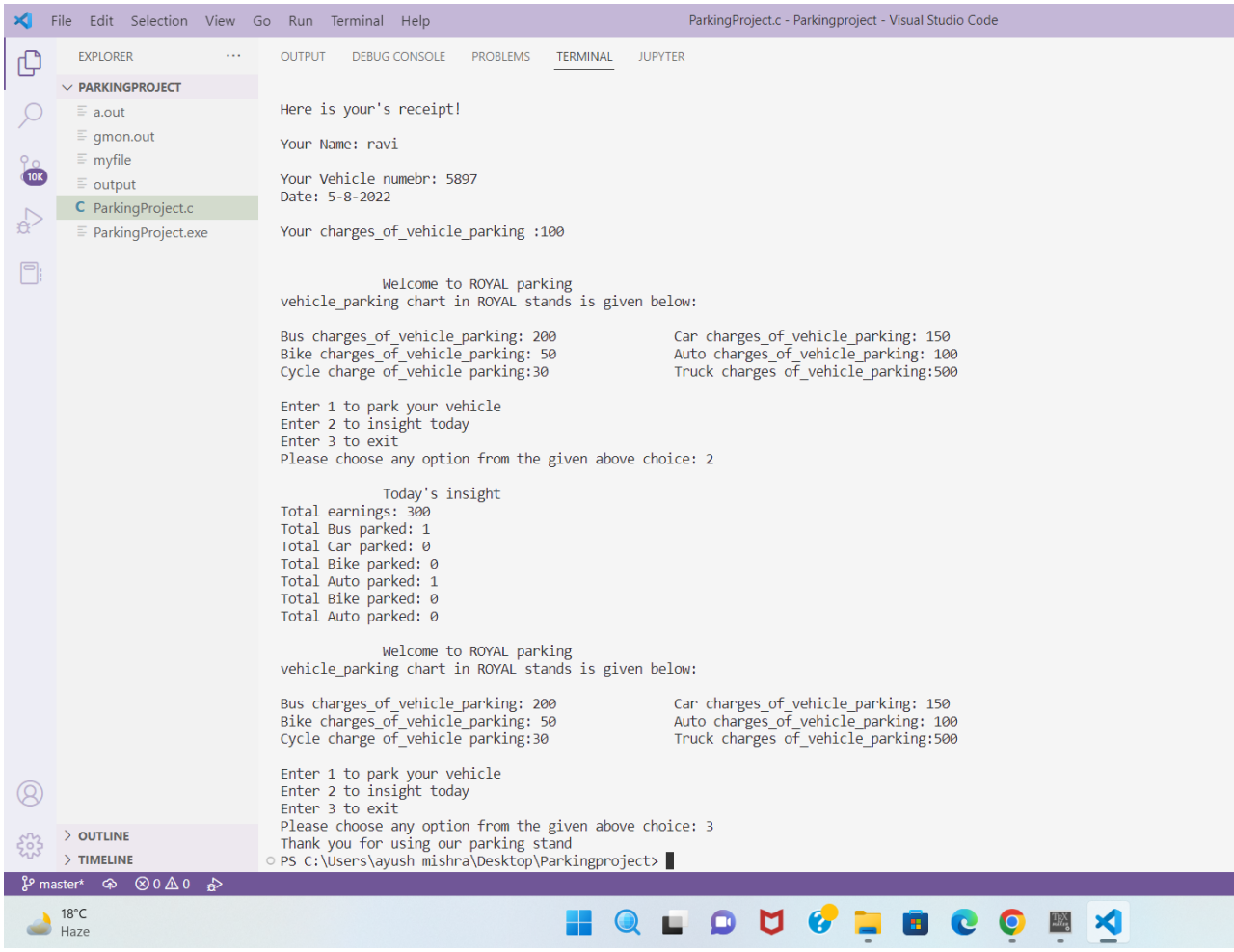
```
        printf("Invalid number try again\n");
        goto again;
        exit(0);
    }
}

// This function use to check parking insight
void check_insight()
{
    printf("\n              Today's insight               \n");
    printf("Total earnings: %d\n", earnings);
    printf("Total Bus parked: %d\n", BusCount);
    printf("Total Car parked: %d\n", CarCount);
    printf("Total Bike parked: %d\n", BikeCount);
    printf("Total Auto parked: %d\n", AutoCount);
    printf("Total Bike parked: %d\n", CycleCount);
    printf("Total Auto parked: %d\n", TruckCount);
}

// this function comment to customer and then exit out.
void exit_function(){
    printf("Thank you for using our parking stand\n");
}

// This is our main function in the code
int main()
{
    int choose;
    do
    {
        printf("\n              Welcome to ROYAL parking           \n");
        printf("vehicle parking chart in ROYAL stands is given below: \n");
        printf("\n");
        printf("Bus charges of vehicle parking: 200               Car charges of
        printf("Bike charges of vehicle parking: 50               Auto charges o
        printf("Cycle charge of vehicle parking:30                Truck charges
        printf("\nEnter 1 to park your vehicle\n");
        printf("Enter 2 to insight today\n");
        printf("Enter 3 to exit\n");
    again:
        printf("Please choose any option from the given above choice: ");
        scanf("%d", &choose);
        switch (choose)
        {
        case 1:
            park_vehicle();
```

```
            break;
        case 2:
            check_insight();
            break;
        case 3:
            exit_function();
            exit(0);
            break;
        default:
            printf("Invalid number try again\n");
            goto again;

            break;
        }
    } while (choose != 4);
}
```

CODE OUTPUT


File Edit Selection View Go Run Terminal Help
ParkingProject.c - Parkingproject - Visual Studio Code
EXPLORER
PARKINGPROJECT
a.out
gmon.out
myfile
output
ParkingProject.c
ParkingProject.exe
OUTLINE
TIMELINE
OUTPUT DEBUG CONSOLE PROBLEMS TERMINAL JUPYTER
PS C:\Users\ayush mishra\Desktop\Parkingproject> cd "c:\Users\ayush mishra\Desktop\Parkingproject\" ; if ($?) { gcc Parkingproject }
Welcome to ROYAL parking
vehicle_parking chart in ROYAL stands is given below:
Bus charges_of_vehicle_parking: 200
Car charges_of_vehicle_parking: 150
Bike charges_of_vehicle_parking: 50
Auto charges_of_vehicle_parking: 100
Cycle charge of_vehicle parking:30
Truck charges of_vehicle_parking:500
Enter 1 to park your vehicle
Enter 2 to insight today
Enter 3 to exit
Please choose any option from the given above choice: 1
1.Bus
2.Car
3.Bike
4.Auto
5.Cycle
6.Truck
Please tell which vehicle do you want to park: 1
Please Enter your name: ayush
Please Enter vehicle number: 123
Please Enter today's date: 2-5-2002
Please enter charges_of_vehicle_parking: 200
Your payment is successfully done
Here is your's receipt!
Your Name: ayush
Your Vehicle numebr: 123
Date: 2-5-2002
Your charges_of_vehicle_parking :200
Welcome to ROYAL parking
vehicle_parking chart in ROYAL stands is given below:
Bus charges_of_vehicle_parking: 200
Car charges_of_vehicle_parking: 150
Bike charges_of_vehicle_parking: 50
Auto charges_of_vehicle_parking: 100
Cycle charge of_vehicle parking:30
Truck charges of_vehicle_parking:500
Enter 1 to park your vehicle
Enter 2 to insight today
Enter 3 to exit
master*
18°C
Haze


File Edit Selection View Go Run Terminal Help
ParkingProject.c - Parkingproject - Visual Studio Code
EXPLORER
PARKINGPROJECT
a.out
gmon.out
myfile
output
ParkingProject.c
ParkingProject.exe
OUTLINE
TIMELINE
OUTPUT DEBUG CONSOLE PROBLEMS TERMINAL JUPYTER
Enter 3 to exit
Please choose any option from the given above choice: 1
1.Bus 2.Car
3.Bike 4.Auto
5.Cycle 6.Truck
Please tell which vehicle do you want to park: 4
Please Enter your name: ravi
Please Enter vehicle number: 5897
Please Enter today's date: 5-8-2022
Enter parking charges_of_vehicle_parking: 100
Your payment is successfully done
Here is your's receipt!
Your Name: ravi
Your Vehicle numebr: 5897
Date: 5-8-2022
Your charges_of_vehicle_parking :100
Welcome to ROYAL parking
vehicle_parking chart in ROYAL stands is given below:
Bus charges_of_vehicle_parking: 200 Car charges_of_vehicle_parking: 150
Bike charges_of_vehicle_parking: 50 Auto charges_of_vehicle_parking: 100
Cycle charge of_vehicle parking:30 Truck charges of_vehicle_parking:500
Enter 1 to park your vehicle
Enter 2 to insight today
Enter 3 to exit
Please choose any option from the given above choice: 2
Today's insight
Total earnings: 300
Total Bus parked: 1
Total Car parked: 0
Total Bike parked: 0
Total Auto parked: 1
Total Bike parked: 0
Total Auto parked: 0
Welcome to ROYAL parking
master*
18°C
Haze


File Edit Selection View Go Run Terminal Help
ParkingProject.c - Parkingproject - Visual Studio Code
EXPLORER
PARKINGPROJECT
a.out
gmon.out
myfile
output
ParkingProject.c
ParkingProject.exe
OUTLINE
TIMELINE
OUTPUT DEBUG CONSOLE PROBLEMS TERMINAL JUPYTER
Here is your's receipt!
Your Name: ravi
Your Vehicle numebr: 5897
Date: 5-8-2022
Your charges_of_vehicle_parking :100
Welcome to ROYAL parking
vehicle_parking chart in ROYAL stands is given below:
Bus charges_of_vehicle_parking: 200 Car charges_of_vehicle_parking: 150
Bike charges_of_vehicle_parking: 50 Auto charges_of_vehicle_parking: 100
Cycle charge of_vehicle parking:30 Truck charges of_vehicle_parking:500
Enter 1 to park your vehicle
Enter 2 to insight today
Enter 3 to exit
Please choose any option from the given above choice: 2
Today's insight
Total earnings: 300
Total Bus parked: 1
Total Car parked: 0
Total Bike parked: 0
Total Auto parked: 1
Total Bike parked: 0
Total Auto parked: 0
Welcome to ROYAL parking
vehicle_parking chart in ROYAL stands is given below:
Bus charges_of_vehicle_parking: 200 Car charges_of_vehicle_parking: 150
Bike charges_of_vehicle_parking: 50 Auto charges_of_vehicle_parking: 100
Cycle charge of_vehicle parking:30 Truck charges of_vehicle_parking:500
Enter 1 to park your vehicle
Enter 2 to insight today
Enter 3 to exit
Please choose any option from the given above choice: 3
Thank you for using our parking stand
PS C:\Users\ayush mishra\Desktop\Parkingproject>
master*
18°C
Haze