

# Toll Plaza Management System

Mini-Project

15,November 2022

## **1 ABSTRACT:**

Toll Plaza Management system is designed to keep track of the vehicle's movement, record vehicle name etc. This system is very useful for vehicle tracking, time management and also for automation of Toll gate. Online Toll Gate Management systems have been of great assistance in lessening the over congestion that has become a part of the metropolitan cities these days. It is one of the uncomplicated ways to manage the great run of traffic. The travelers passing through this mode of transport, carried by their transport that allows them to be aware of the account of money that has been paid and the money left in the tag. It relieves the traveler of the burden of waiting in the queue to make the toll payment, which decreases the fuel-consumption.

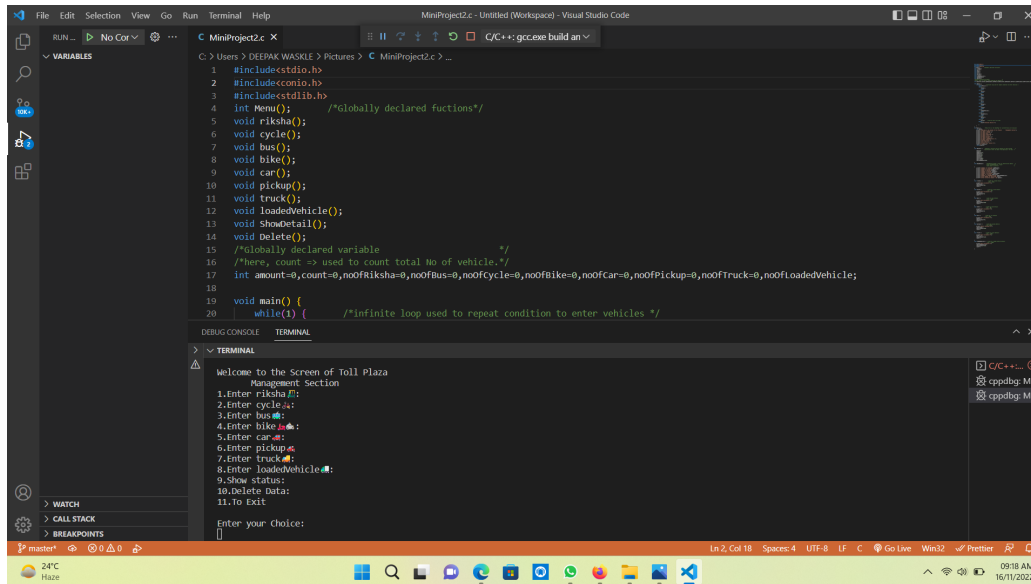
## **2 SYSTEM HOME SCREEN:**

System of Toll Plaza's Home screen where we have different-different option to choose for action which we actually want see figure-1 .

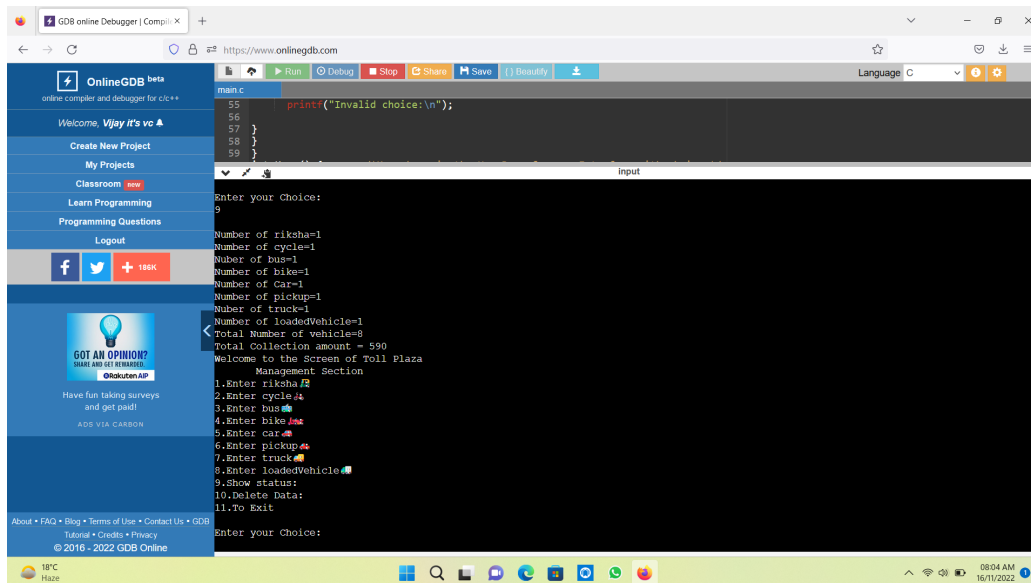
## **3 SYSTEM OUTPUT SCREEN:**

(SEE figure-2)

- A. System of Toll Plaza's output screen where we can see the total number of vehicle including bike and cycle .
- B. But there is no tax for bike and cycle excluding other vehicle .
- C. We can see here total amount collected in whole day.



(a) figure-1



(b) figure-2

## **4 DEBUGGING OUTCOMES:**

A.see figure-3

B.see figure-4

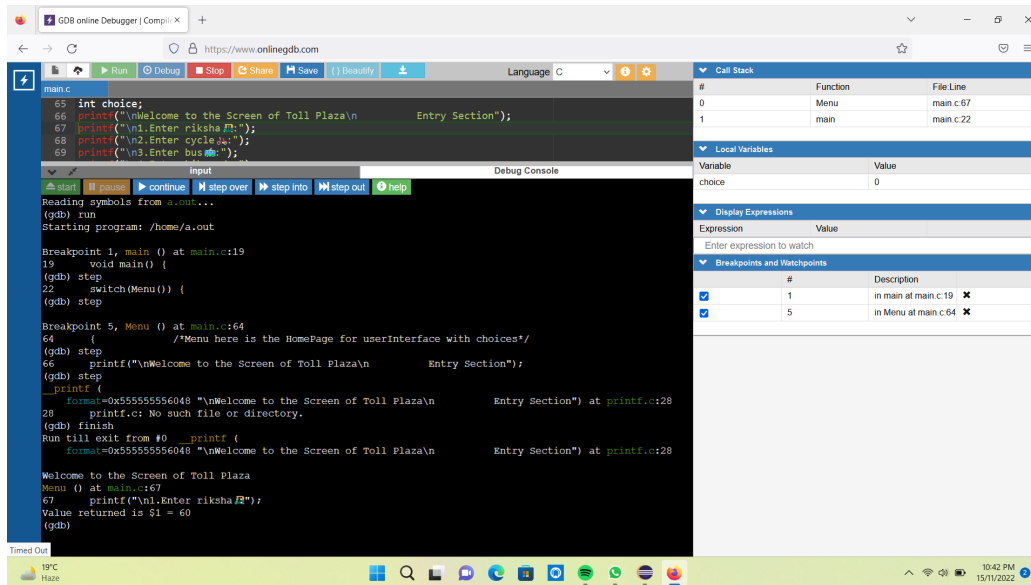
### **4.1 COMMANDS USED TO DEBUG PROGRAM:**

A.see figure-5

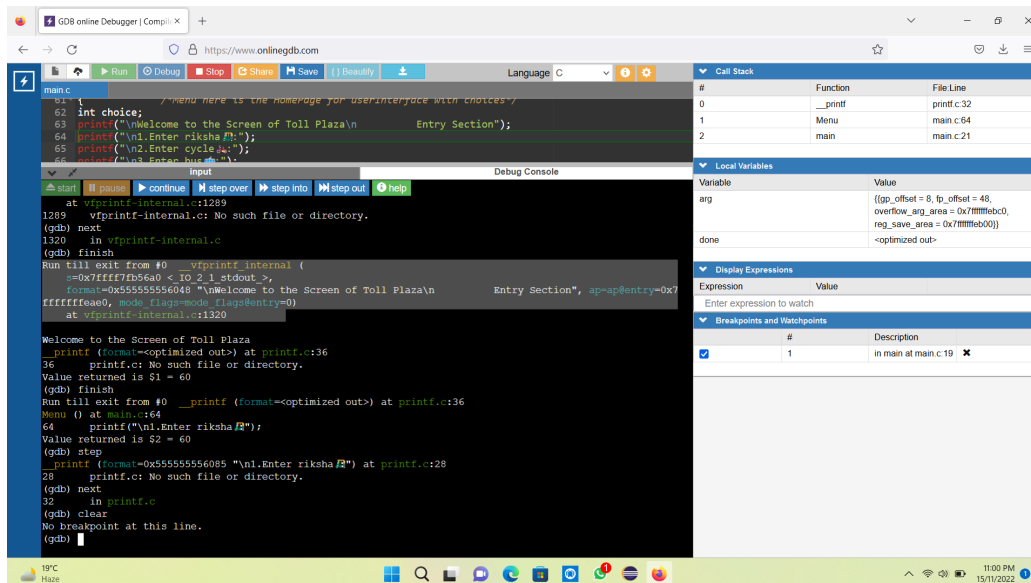
## **5 PROFILING OUTCOMES:**

### **5.1 COMMANDS USED IN PROFILING-**

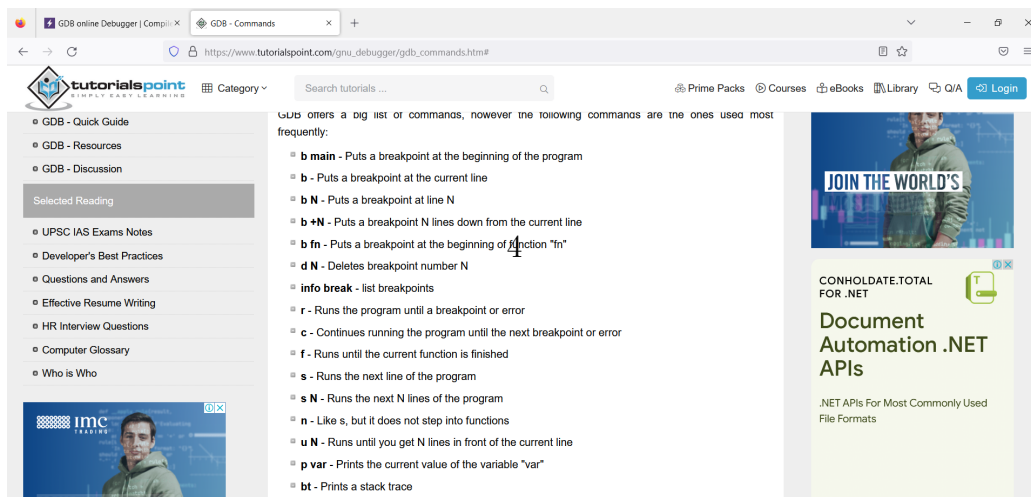
1. `sudo apt-get install binutils`
2. `gcc -Wall -pg vijay.c -o vijay`
3. `./test`
4. `gprof vijay gmon.out |> vijay.txt`



(c) figure-3



(d) figure-4



```

3
Entry success👍👍

Welcome to the Screen of Toll Plaza
Management Section
1.Enter riksha🚲:
2.Enter cycle🚲:
3.Enter bus🚌:
4.Enter bike🏍️:
5.Enter car🚗:
6.Enter pickup🚚:
7.Enter truck🚚:
8.Enter loadedVehicle🚚:
9.Show status:
10.Delete Data:
11.To Exit

Enter your Choice:
^Z
[2]+ Stopped ./vijay
monu@Charlie:~$ ls
TICTACTOE TICTACTOE.c TICTACTOE.txt a.out conio.h gmon.out vijay vijay.c
monu@Charlie:~$ gprof vijay gmon.out>vijay.txt
monu@Charlie:~$ gprof vijay a.out>vijay.txt
gprof: file 'a.out' does not appear to be in gmon.out format
monu@Charlie:~$ ls
TICTACTOE TICTACTOE.c TICTACTOE.txt a.out conio.h gmon.out vijay vijay.c vijay.txt
monu@Charlie:~$ gprof vijay gmon.out>vijay.txt
monu@Charlie:~$ cat vijay.txt
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 6 0.00 0.00 cycle
0.00 0.00 0.00 5 0.00 0.00 Delete

% the percentage of the total running time of the

```

(f) figure-6

```

monu@Charlie:~$ gprof vijay gmon.out>vijay.txt
monu@Charlie:~$ cat vijay.txt
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 6 0.00 0.00 cycle
0.00 0.00 0.00 5 0.00 0.00 Delete

% 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.

```

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     | 1/6    | Menu [3]   |
|       |        | 0.00 | 0.00     | 5/6    | main [14]  |
| [1]   | 0.0    | 0.00 | 0.00     | 6      | cycle [1]  |
| ----- |        |      |          |        |            |
|       |        | 0.00 | 0.00     | 5/5    | Menu [3]   |
| [2]   | 0.0    | 0.00 | 0.00     | 5      | Delete [2] |
| ----- |        |      |          |        |            |

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%.

Figure 1: call graph

## **6 ADVANTAGE OF PROPOSED SYSTEM:**

1. Gives accurate information.
2. Simplifies the manual work.
3. Gives the correct calculation of amount and vehicle.
4. Home page is user friendly.

## **7 DISADVANTAGE OF PROPOSED SYSTEM:**

1. In this system each schedule can be tracked from the start till the end of the Project cycle.
2. User friendliness is provided in the application with various controls.
3. The system makes the overall project management much easier and flexible.
4. There is no risk of data mismanagement at any level while the project is being running.

## **8 SYSTEM REQUIREMENTS:**

### **HARDWARE REQUIREMENTS:**

- a. System : Any with Dual Core.
- b. Hard Disk : not much .
- c. Input Devices : Keyboard, Mouse
- d. Ram : 1 GB

### **SOFTWARE REQUIREMENTS:**

- a. Operating system : window-7,8,9,11
- b. Coding Language : C#.net
- c. Tool : Visual Studio

## 9 CODE IN C:

```
#include<stdio.h >
#include<conio.h >
#include<stdlib.h>
int Menu (); /*Globally declared fuctions */
void riksha ();
void cycle ();
void bus ();
void bike ();
void car ();
void pickup ();
void truck ();
void loadedVehicle ();
void ShowDetail ();
void Delete ();
/*Globally declared variable */
/*here, count =j used to count total No of vehicle*/
int amount = 0, count = 0, noOfRiksha = 0, noOfBus = 0, noOfCycle = 0,
noOfBike = 0, noOfCar = 0, noOfPickup = 0, noOfTruck = 0, noOfLoaded-
Vehicle;

void main () //main fuction
{
while (1)
{ /*infinite loop used to repeat condition to enter vehicles */
switch (Menu ())
{
case 1:
riksha ();
break;
case 2:
cycle ();
break;
case 3:
bus ();
break;
case 4:
```



```

    bike ();
    break;
    case 5:
    car ();
    break;
    case 6:
    pickup ();
    break;
    case 7:
    truck ();
    break;
    case 8:
    loadedVehicle ();
    break;
    case 9:
    ShowDetail ();
    break;
    case 10:
    Delete ();
    break;
    case 11:
    exit (0); //used to exit from loop.
    default:
    printf ("Invalid choice:");

    }
}

```

```

    int Menu ()
{ /*Menu here is the HomePage for userInterface with choices */
int choice;
printf ("to the Screen of Toll PlazaEntry Section");
printf ("1.Enter rikshap:");
printf ("2.Enter cyclep2:");
printf ("3.Enter busp:");
printf ("4.Enter bikep5po8:");
printf ("5.Enter carp:");
printf ("6.Enter pickupp;");

```

```

printf ("7.Enter truckp:");
printf ("8.Enter loadedVehiclep:");
printf ("9.Show status:");
printf ("10.Delete Data:");
printf ("11.To Exit");
printf ("your Choice:");
scanf ("return (choice);

}

void Delete ()
{ /*Delete=, function used to delete all data entered */
/*and Restart Enter new data from begining of the day */
amount = 0;
count = 0; noOfRiksha = 0;
noOfBus = 0; noOfCycle = 0;
noOfBike = 0; noOfCar = 0;
noOfPickup = 0; noOfTruck = 0;
noOfLoadedVehicle = 0;
}
void ShowDetail () /*ShowDetail=, Used to show all Entered data Datal */
{ /*and totalNoOfVehicle =count */
/*and amount collection */
printf ("of riksha=printf ("of cycle=printf ("of bus=printf ("of bike=printf
("of Car=printf ("of pickup=printf ("of truck=printf ("of loadedVehicle=printf
("Number of vehicle=printf ("Collection amount =
}
void riksha ()
{ //used for riksha details
//tax = 50Rs.
printf ("Entry successbp");
noOfRiksha++;
amount = amount + 50;
count++;

}

void cycle ()

```

```

{ //used for cycle details
//tax = 0Rs.
printf ("Entry successbp");
noOfCycle++;
amount = amount + 0;
count++;

}

void bus ()
{ //used for bus details
//tax = 100Rs.
printf ("Entry successbp");
noOfBus++;
amount = amount + 100;
count++;

}

void bike ()
{ //used for bike details
//tax = 0Rs.
printf ("Entry successbp");
noOfBike++;
amount = amount + 0;
count++;

}

void car ()
{ //used for car details
//tax = 70Rs.
printf ("Entry successbp");
noOfCar++;
amount = amount + 70;
count++;

}

```

```

        void pickup ()
        { //used for pickup details
        //tax =100Rs.
        printf ("nEntry successbp");
        noOfPickup++;
        amount = amount + 100;
        count++;

        }

        void truck ()
        { //used for truck deatails
        //tax = 120Rs.
        printf ("Entry successbp");
        noOfTruck++;
        amount = amount + 120;
        count++;

        }

        void loadedVehicle ()
        { //used for loaded vehicle details
        //tax = 150Rs.
        printf ("Entry successbp");
        noOfLoadedVehicle++;
        amount = amount + 150;
        count++;

        }

```

## 10 CODE IN JAVA:

```

import java.util.*;
class MiniProject
{

        /*Globally declared variable */

```

```

/*here, count =i used to count total No of vehicle. */
int amount = 0, count = 0, noOfRiksha = 0, noOfBus = 0, noOfCycle = 0,
noOfBike = 0, noOfCar = 0, noOfPickup = 0, noOfTruck = 0, noOfLoaded-
Vehicle;

    public int Menu ()
    {
/*Menu here is the HomePage for userInterface with choices */
System.out.println ("to the Screen of Toll Plaza");
System.out.println (" Management Section");
System.out.println ("1.Enter rikshap:");
System.out.println ("2.Enter cyclep2:");
System.out.println ("3.Enter busp:");
System.out.println ("4.Enter bikep5po8:");
System.out.println ("5.Enter carp:");
System.out.println ("6.Enter pickupp:");
System.out.println ("7.Enter truckp:");
System.out.println ("8.Enter loadedVehiclep:");
System.out.println ("9.Show status:");
System.out.println ("10.Delete Data:");
System.out.println ("11.To Exit");
System.out.println ("your Choice:");
Scanner sc = new Scanner (System.in);
int choice = sc.nextInt ();
return (choice);

    }
    public void Delete ()
    { /*Delete=i function used to delete all data entered */
/*and Restart Enter new data from begining of the day */
amount = 0;
count = 0; noOfRiksha = 0;
noOfBus = 0; noOfCycle = 0;
noOfBike = 0; noOfCar = 0;
noOfPickup = 0; noOfTruck = 0;
noOfLoadedVehicle = 0;
    }
    public void ShowDetail ()

```

```

{ /*ShowDetail=Used to show all Entered data Datal */
/*and totalNoOfVehicle =count */
/*and amount collection */
System.out.print ("of riksha=" + noOfRiksha);
System.out.print ("of cycle=" + noOfCycle);
System.out.print ("of bus=" + noOfBus);
System.out.print ("of bike=" + noOfBike);
System.out.print ("of Car=" + noOfCar);
System.out.print ("of pickup=" + noOfPickup);
System.out.print ("of truck=" + noOfTruck);
System.out.print ("of loadedVehicle=" + noOfLoadedVehicle);
System.out.print ("Number of vehicle=" + count);
System.out.print ("Collection amount = " + amount);

}
public void riksha ()
{ //used for riksha details
//tax = 50Rs.
System.out.print ("Entry successbp");
noOfRiksha++;
amount = amount + 50;
count++;

}
public void cycle ()
{ //used for cycle details
//tax = 0Rs. System.out.print ("Entry successbp");
noOfCycle++;
amount = amount + 0;
count++;

}
public void bus ()
{ //used for bus details
//tax = 100Rs.
System.out.print ("Entry successbp");
noOfBus++;
amount = amount + 100;

```

```

count++;

    }
    public void bike ()
    { //used for bike details
    //tax = 0Rs.
    System.out.print ("Entry successbp");
    noOfBike++;
    amount = amount + 0;
    count++;

    }
    public void car ()
    { //used for car details
    //tax = 70Rs.
    System.out.print ("Entry successbp");
    noOfCar++;
    amount = amount + 70;
    count++;

    }
    public void pickup ()
    { //used for pickup details
    //tax =100Rs.
    System.out.print ("nEntry successbp");
    noOfPickup++;
    amount = amount + 100;
    count++;

    }
    public void truck ()
    { //used for truck deatails
    //tax = 120Rs.
    System.out.print ("Entry successbp");
    noOfTruck++;
    amount = amount + 120;
    count++;

```

```

    }
    public void loadedVehicle ()
    { //used for loaded vehicle details
    //tax = 150Rs.
    System.out.print ("Entry successbp");
    noOfLoadedVehicle++;
    amount = amount + 150;
    count++;

    }
}

class Main
{

    public static void main (String args[])
    {

        //Indtsntiated an object of class MiniProject.
        MiniProject mini = new MiniProject ();
        while (1 != 0)
        { /*infinite loop used to repeat condition to enter vehicles */
        switch (mini.Menu())
        {
        case 1:
        mini.riksha ();
        break;
        case 2:mini.cycle ();
        break;
        case 3:mini.bus ();
        break;
        case 4:mini.bike ();
        break;
        case 5:mini.car ();
        break;
        case 6:mini.pickup ();
        break;
        case 7:mini.truck ();

```



```
break;
case 8:mini.loadedVehicle ();
break;
case 9:mini.ShowDetail ();
break;
case 10:mini.Delete ();
break;
case 11:System.exit (0); //used to exit from loop.
default:System.out.print ("Invalid choice.");

    }

}

}
```