**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter No** | **Chapter Names** | **Page No** |
| **1** | **INTRODUCTION** | **1** |
| **2** | **SYSTEM REQUIREMENTS** | **2** |
| **3** | **PROBLEM STATEMENT, OBJECTIVES AND SCOPE** | **3** |
| **4** | **DEFINTION OF PROBLEM** | **4** |
| **5** | **EXISTING SYSTEM** | **5** |
| **6** | **TECHNOLOGY USED** | **6** |
| **7** | **SOURCE CODE** | **8** |
| **8** | **OUTPUT** | **10** |
| **9** | **LIMITATION AND FUTURE ENHANCEMENT** | **17** |
| **10** | **CONCLUSION** | **21** |
| **11** | **REFERENCES** | 23 |

**CHAPTER 1**

**INTRODUCTION**

Parking management system for managing the records of the incoming and outgoing vehicles in an parking house It's an easy for Admin to retrieve the data if the vehicle has been visited through number he can get that data.

Now days in many public places such as malls, multiplex system, hospitals, offices, market areas there is a crucial problem of vehicle parking. The vehicle parking area has many lanes/slots for car parking. So to park a vehicle one has to look for all the lanes. Moreover this involves a lot of manual labour and investment. Instead of vehicle caught in towing the vehicle can park on safe and security with low cost.

Parking control system has been generated in such a way that it is filled with many secure devices such as, parking control gates, toll gates, time and attendance machine, car counting system etc. These features are hereby very necessary nowadays to secure your car and also to evaluate the fee structure for every vehicles entry and exit.

The objective of this project is to build a Vehicle Parking management system that enables the time management and control of vehicles using number plate recognition. The system that will track the entry and exit of cars, maintain a listing of cars within the parking lot, and determine if the parking lot is full or not. It will determine the cost of per vehicle according to their time consumption.Our project helps to calculate how many cars have been rented in the parking lot and how many vehicles and what kind of vehicles are there and what kind of vehicles are there is still in the parking lot and how many vehicles have arrived and the parking will be full. This created in such way that helps the user to do the management of the vehicles.

**Used header files**

1. **iostream :** iostream stands for standard input-output stream. This header file contains definition to objects like cin, cout etc.
2. **process.h :** is the header file in c++ that contains functional declaration and macros.
3. **stdlib.h :** is the header of the general purpose standard library of c programming language which include functions involving memory allocation, process control, conversion and others. it is compatible with c++ and is known as stdlib in c++. The name “stdlib” stands for “standard library”.

**Features Used of C++**

1. **Classes And Object**

A class in C++ is building blocks, that leads to object oriented programming. it is a user- define data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class.

An Object is an instance of a class.

1. **Inheritance**

the capability of a class to derive properties and characteristics from another class is called inheritance.

1. **Constructor**

A constructor is a member function of a class which initialises objects of a class.

**CHAPTER 2**

**SYSTEM REQUIREMENTS**

|  |  |
| --- | --- |
| RAM | 2 GB OR HIGHER |
| MONITOR | 17 INCH COLOUR MONITOR |
| KEYBORD | 104 KEYBORD |
| MOUSE | STANDARD MOUSE |
| PROCESSOR | Intel i3 OR HIGHER |

HAREDWARE

SOFTWARE

|  |  |
| --- | --- |
| OPERATING SYSTEM | WINDOWS 7 OR HIGHER |
| PROGRAM | TURBO C++ COMPUTER / CODE::BLOCKS |

**CHAPTER 3**

**PROBLEM STATEMENT, OBJECTIVE AND SCOPE**

New road parking had challenges concerning its safety of data in the store since they currently use paper based system, physical struggle for parking by drivers, wastage of time, congregation and collision. They was also a problem of monitoring the profit made for company where by the company was losing money to its workers who received the money. This is system majorly solved the congestion, collision and save time during parking activities.

# Objectives :

To enable drivers to locate and reserve a parking place online through accessing it on web platform.

We can park a vehicle in our vehicle in our own slot by paying. Because of that there is no towing problems.

And our vehicle has been parked as a secure condition. There is no risk for vehicle owner for parking the car.

In case of any damages and problem of vehicle that will claim by parking management.

As the world is facing many threads daily, robberies are done easily with no track to trace, bomb blasts occur with the use of vehicles, so if a proper system is adopted each and every record can be saved and anyone can be track easily there fore mainly is to make a better and fast software, most important user-friendly

Maintain records in short time of period. Determines the parking area is full or not. Enhances the visitor’s experience.

# scope :

Can be easily used in public parking lot.

The congestion and collision of vehicle due to previous manual method will be little to no longer exist.

In the modern age. Many people have vehicles. Vehicle is now a basic need. Every place is under a process of urbanization. There are many corporate offices and shopping centers etc. there are many recreational places where people use to go for refreshment. So, all these places need a parking space where people can park their vehicles safely and easily.

Every parking or need a system that records and the details of vehicles to give the facility. These system might be computerized or non-computerized. with the help of computerization system we can deliver a good service to customer who wants to park their vehicle into any organizations premises.

Vehicle parking management system is an automatic system which delivers data processing in very high speed in systematic manner. Parking is a growing need of the time. Development of this system is very useful in this area of field. We can sell this system to any organization. By using our system they can maintain records very easily. Our system covers the every area of parking management.

In coming future there will be excessive need of vehicle parking management system.

**CHAPTER 4**

**DEFINITION OF PROBLEM** :

Now a days in parking like valet parking they maintain just with the tokens and they have records the vehicle details in books so that during some critical situation like this enquiry of terrorist can or vehicle borer are that case it is difficult to find that details of particular vehicle but in this case is it to find in 1 to 2 seconds.

By parking the vehicle in public place the vehicle can be claimed by towing person but in this case there is no towing problem and no need to give fine for anything we can park our vehicle with security.

The problem of vehicle parking system refers to the challenges and issues associated with the efficient management and organization of parking spaces for vehicles in a particular area or facility. This problem arises due to the limited availability of parking spots compared to the number of vehicles that require parking.

The problem can manifest in various ways, including:

* 1. Insufficient parking spaces: In areas with high population density or limited infrastructure, there may not be enough parking spots to accommodate all the vehicles, leading to a shortage of available parking spaces.
  2. Inefficient space utilization: Even if there are an adequate number of parking spaces, they may not be utilized efficiently, resulting in inefficient use of the available area. This can occur due to improper parking behavior, lack of clear signage or markings, or poor parking management systems.
  3. Traffic congestion: Poorly managed parking systems can lead to traffic congestion in and around parking areas, especially during peak hours. Vehicles circling around in search of parking spots can contribute to traffic congestion and increase the overall commute time.
  4. Unauthorized parking: Unauthorized parking, such as parking in designated spots for disabled individuals or in restricted areas, can create problems and inconvenience for others. It can also lead to safety hazards or block emergency access routes.
  5. Lack of information and guidance: Inadequate information and guidance about available parking spaces, such as real-time occupancy information or clear directions to parking areas, can make it challenging for drivers to find suitable parking spots efficiently.

Addressing the problem of vehicle parking system involves implementing effective strategies and solutions, including:

* Developing comprehensive parking management plans that consider factors such as demand, space allocation, and enforcement.
* Utilizing technologies such as parking sensors, smart parking meters, and mobile applications to provide real-time parking availability information to drivers.
* Implementing efficient parking layout designs, clear signage, and markings to optimize space utilization and guide drivers.
* Enforcing parking regulations and penalties for unauthorized parking to ensure compliance.
* Promoting alternative modes of transportation, such as public transit or carpooling, to reduce the demand for parking spaces.
* Investing in infrastructure expansion or construction of multi-level parking facilities to increase parking capacity in high-demand areas.

By addressing the challenges associated with the vehicle parking system, it is possible to enhance the overall parking experience, reduce traffic congestion, improve safety, and optimize the use of available parking resources.problem.

**CHAPTER 5**

**EXISTING SYSTEM :**

In the modern age. Many people have vehicles. The vehicle is now a basic need. Every place is under the process of urbanization. There are many corporate offices and shopping centers etc. There are many recreational places where people used to go for a refreshment. So, all these places need a parking space where people can park their vehicles safely and easily.

Every parking area needs a system that records the detail of vehicles to give the facility.

These systems might be computerized or non-computerized. With the help of the computerized system, we can deliver a good service to the customer who wants to park their vehicle into any organization's premises.

PROPOSED SYSTEM:

Nowadays in parking like valet parking they maintain just with the tokens and they record the vehicle details in books so, during some critical situations like police inquiry for terrorist's car or vehicle, it is difficult to find the details of the particular vehicle but in this case, it is easy to find within few seconds only.

By parking the vehicle in public place the vehicle can be claimed by towing person but in this case, there are no towing problems and no need to give fine for anything we can park our vehicle with security.

The vehicle parking management system is an automatic system that delivers data processing at a very high speed in a systematic manner. Parking is a growing need of the time. The development of this system is very useful in this area of the field. We can sell this system to any organization. By using our system they can maintain records very easily. Our system covers every area of parking management. In the coming future, there will be an excessive need for a Vehicle parking management system.

Vehicle Records: - This most important record which focuses in our Vehicle Parking Management System. It stores the essential Vehicle records like: vehicle type Transaction Detail:-This report will show the Transaction between the customer and the System.. It shows the cost of the vehicle after using the facility of parking.

MODULES:

1. Data Records:

Vehicle Records: This most important record which focuses in our Vehicle Parking Management System. It stores the essential Vehicle records like: vehicle type Transaction Detail:-This report will show the Transaction between the customer and the System.. It shows the cost of the vehicle after using the facility of parking.

Parking control system has been generated in such a way that it is filled with many secure devices such as, parking control gates, toll gates, time and attendance machine, car contenting system etc. These features are here by very necessary nobody to secure your car and also to evaluate the fee structure for every vehicle entry and exit.

The objective of this object project is to build a vehicle parking management system that enable to the time management and control of vehicle using number plate recognition. the system that will track the entry and exit of cars, maintain a list of car within a parking lot, and determine if the parking lot is full or not. it will determine the cost of per vehicle according to their time consumption. Our project help to calculate how many car have been rented in the parking lot and how many vehicles and what kind of vehicle are there and what kind of vehicle are there is still in parking lot and how many vehicle how arrived and the parking will be full. This created in such a way that helps the user to do the management of the vehicles.

**CHAPTER 6**

**TECHNOLOGY USED**

# C++

C++ is standardized by the International Organization for Standardization (IDE) with the latest standard version ratified and published by ISO in December 2017 as ISO/IEC 14882:2017 (information known as C++ 17). The C++ programming language was initially standardized in 1998 as ISO/IEC 14882:1998, which was then amended by the C++ 03, C++ 11 and C++ 14 standards.

C++ is a high-level language that has developed in mid of 1970's. It is built or derived from C language. It has object oriented features which allows programmer to create objects within the code. This creation of objects makes programming easier, efficient in terms of time and space. C++ offers power and flexibility over the language.

C++ is a general purpose programming language and widely used now a days for competitive programming. It has imperative, object- oriented and generic programming features. C++ runs on lots of platform like Windows, Linux, Unix, Mac etc.

> C++ Classes and Objects

# Class

The building block of C++ that leads to Object Oriented programming is a Class. It is a user defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A class is like a blueprint for an object.

When you define a class, you define a blueprint for a data type. This doesn't actually define any data, but it does define what the class name means, that is, what an object of the class will consist of and what operations can be performed on such an object A class definition starts with the keyword class followed by the class name; and the class body. enclosed by a pair of curly braces. A class definition must be followed either by a semicolon or a list of declarations

The keyword public determines the access attributes of the members of the class that follows it A public member can be accessed from outside the class anywhere within the scope of the class object. You can also specify the members of a class as private or protected which we will discuss in a sub-section

# Object

Object is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (ie, an object is created) memory is allocated. A class provides the blueprints for objects, so basically an object is created from a class.

We declare objects of a class with exactly the same sort of declaration that we declare variables of basic types

# C++ FUNCTIONS

A function is a group of statements that together perform a task. Every C++ program has at least one function, which is main(), and all the most trivial programs can define additional functions.

You can divide up your code into separate functions. How you divide up your code among different functions is up to you, but logically the division usually is such that each function performs a specific task.

A function declaration tells the compiler about a function's name, return type, and parameters. A function definition provides the actual body of the function. The C++ standard library provides numerous built-in functions that your program can call.

A C++ function definition consists of a function header and a function body. Here are all the parts of a function -

Return Type-A function may return a value. The return type is the data type of the value the function returns. Some functions perform the desired operations without returning a value. In this case, the return\_type is the keyword void.

Function Name - This is the actual name of the function. The function name and the parameter list together constitute the function signature.

Parameters - A parameter is like a placeholder. When a function is invoked, you pass a value to the parameter. This value is referred to as actual parameter or argument The parameter list refers to the type, order, and number of the parameters of a function. Parameters are optional, that is, a function may contains no parameters.

Function Body - The function body contains a collection of statements that define what the function does

# FILE HANDLING IN C++

**Opening a File**

A file must be opened before you can read from it or write to it. Either of stream or fstream object may be used to open a file for writing And ifstream object is used to open a file for reading purpose only.

Following is the standard syntax for open() function, which is a member of fstream, ifstream, und ofstream objects.

# Closing a File

When a C++ program terminates it automatically flushes all the streams, release all the allocated memory and close all the opened files. But it is always a good practice that a programmer should close all the opened files before program termination.

Writing to a File

While doing C++ programming, you write information to a file from your program using the stream insertion operator (c) just as you use that operator to output information to the screen The only difference is that you use an ofstream or fstream object instead of the cout object.

Reading from a File

You read information from a file into your program using the stream extraction operator (>>) Just as you use that operator to input information from the keyboard. The only difference is that you use an ifstream or stream object instead of the cin object.

IMPLEMENTATION PROBLEMSTATEMENT WORKING PROCEDURE

STEP 1

Main function is executed first and the control to the followed by code.

The function "intro" got executed and the welcome message is displayed and when the user proceed to the next step by pressing any key

STEP 2

User prompted with a menu driven screen here and a user with valid input can have the access to the software.

STEP 3

After the user with the valid input logged in, the data of previous vehicle parked records are retrieved

STEP 4

After the user with the valid inputs inserted and successful retrieval, the user land on the home screen. (the Main menu), and based on the choice of the user he land on the requested screen.

STEP 5

After the job done, all the modified or created data is stored in the files and the software is successfully exited.

**CHAPTER 7**

**SOURCE CODE**

#include<iostream> #include<process.h> #include<stdlib.h> using namespace std;

class Parking

{

protected:

int amount; int Count;

int two, three,four; public:

Parking0

{

amount=0; Count=0; two=0; three=0; four-0;

}

void Delet()

{

amount=0; Count=0; two= 0; three=0; four=0;

cout<<’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl; cout<<”Record deleted "<<endl;

cout<<’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

}

};

class Two Wheel:virtual public Parking

{

public:

void t\_wheel()

{

if(Count--50)

{

two++;

amount amount+100;

Count-Count+1; cout<<’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"A new Two Wheeler has parked"<<endl;

cout<<"Total numbers of vehicles "<<Count<<endl;

cout<<"now free space"<<50-(two-three+four)<<endl;

cout<<’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

}

else

{

cout<<"Sorry! parking is full"<<endl;

}

}

};

class Three Wheel:virtual public Parking

{

public:

void th\_wheel()

{

if(Count--50)

{

three++;

amount amount+200; Count-Count+1;

cout<<’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl; cout<<"A new Three Wheeler has parked"<<endl;

cout<<"Total numbers of vehicles "<<Count<<endl; cout<<"now free space"<<50-(two+three+four)<<endl;

cout<<’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

}

else

{

cout<<"Sorry! parking is full";

}

}

}:

class Four Wheel:virtual public Parking public:

void f\_wheel()

{

if(Count = 50)

{

four++; amount=amount+300; Count=Count+1;

cout<<’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"A new Four Wheeler has parked"<<endl;

cout<<"Total numbers of vehicles "<<Count<<endl;

cout<<"now free space="<<50-(two-three+four)<<endl;

cout<<’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

}

else

{

cout<<"Sorry! parking is full";

}

}

};

class ShowRecord:public TwoWheel,public ThreeWheel,public FourWheel

{

public:

void Show

{

cout<<’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl; cout<<"Total amount="<<amount<<endl;

cout<<"Total numbers of vehicles :"<<Count<<endl; cout<<"Total numbers of Two Wheeler :"<<two<<endl; cout<<"Total number of Three Wheeler :"<<three<<endl; cout<<"Total number of Four Wheeler :"<<<<four<<<endl;

cout<<’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

}

};

int main()

{

int u\_input, ShowRecords;

cout<<’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl; cout<<"VEHICLE PARKING SYSTEM"<<endl;

cout<<’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl; cout<<"MADE BY:"<<endl;

cout<<"17. G.Vaishnavi\n";

cout<<"18. Isharani\n";

cout<<’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl; cout<<"Parking number of vehicles limit=50"<<endl;

cout<<"Two Wheeler parking charges\t=100"<<endl; cout<<"Three Wheeler parking charges\t=200"<<endl; cout<<"four Wheeler parking charges\t-300"<<endl;

cout<<’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl; while(1)

{

//Menu

cout<<"Choose our service "<<endl;

cout<<"press 1 for Two Wheeler"<<endl; cout<<"press 2 for Three Wheeler"<<endl; cout<<"press 3 for Four Wheeler"<<endl; cout<<"press 4 for show the record"<<endl; cout<<"press 5 for delete the record"<<endl; cout<<"press 6 for Exit"<<endl;

cout<<’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cin>>u\_input; switch(u\_input)

{

case 1:

system("cls"),

s.t\_wheel()); break;

case 2:

system("cls"); s.th\_wheel() break;

case 3:

system("cls");

s.f\_wheel()); break;

case 4:

system("cls");

s.Show(); break;

case 5:

system("cls");

s.Delet(); break;

Case 6:

exit(0);

default:

{

system("cls"); cout<<’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"WRONGINPUT"<<endl; cout<<’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

}

}

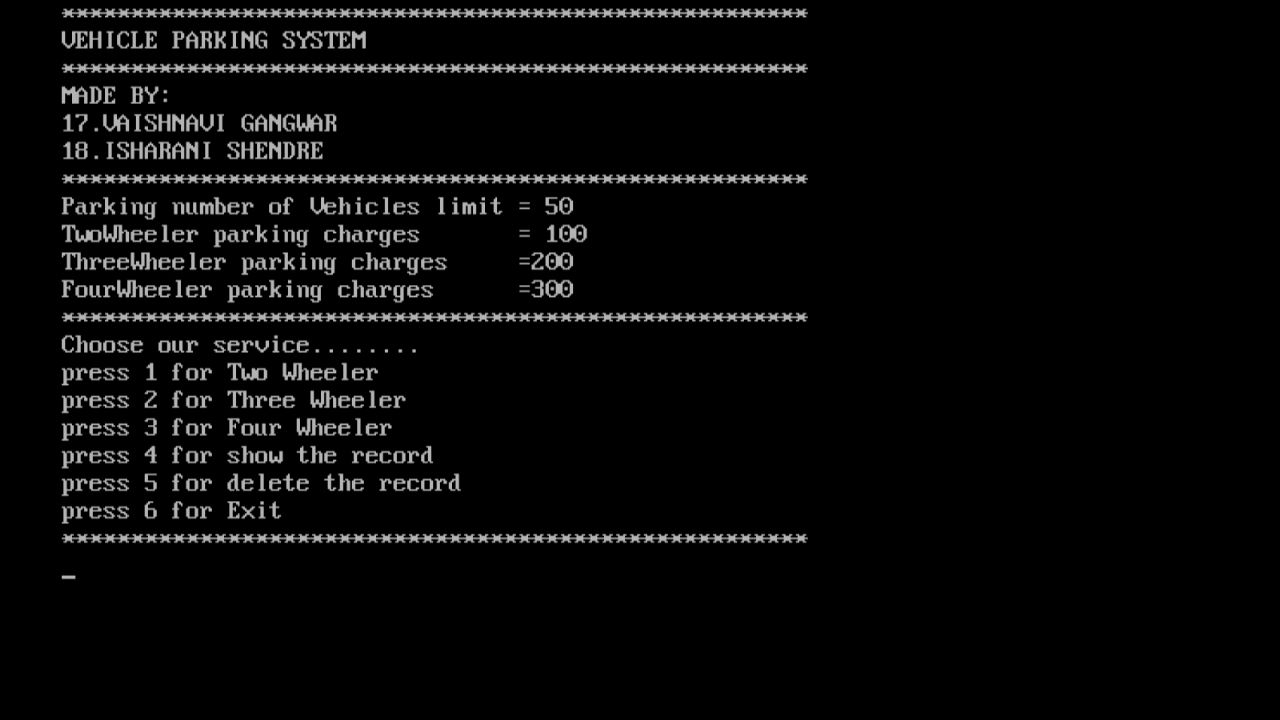
}

return(0);

}

**CHAPTER 8**

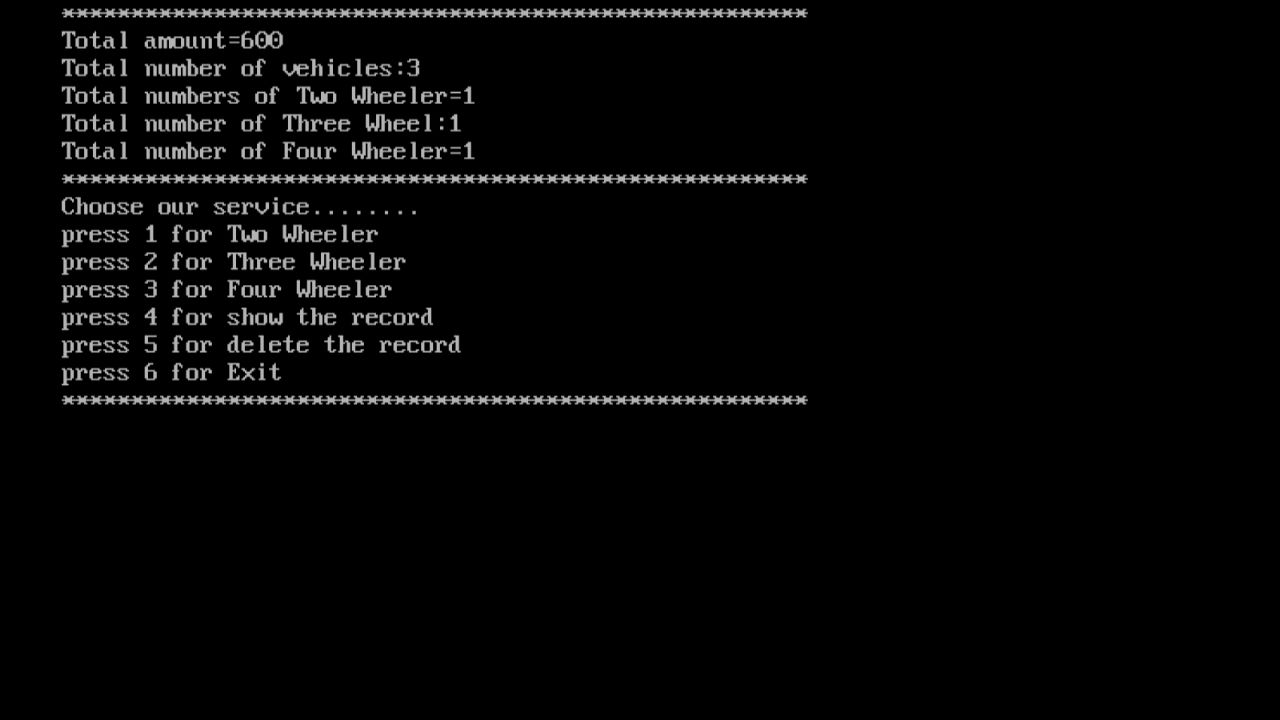
**OUTPUT**



****

****



****

****

**CHAPTER 9**

**LIMITATIONS AND FUTURE ENHANCEMENT**

|  |  |
| --- | --- |
| **LIMITATIONS** | **FUTURE ENHANCEMENT** |
| We can not delete individual vehicle records in the project | In the future, we will be able to delete individual vehicle records in the project |
| You can not calculate rent for more then one day | In future, we calculate more than one day’s rent in a project |
| You can not add a name to our project | In future, you can add your name to the project |

**CHAPTER 10**

**CONCLUSEION**

* This project is created in such a way that helps the user to do the management of rented vehicles and parking of vehicles. This project helps in parking managements. We learn from this project how to create class and objects and also a inheritance, Constructor etc.
* This Project is minimizing the task of parking a vehicle by paying and saying some details about customer and vehicle to save data .In this the vehicle is parked as a safe and secure.
* This project is done as Efficient as possible.
* Hereby I concludes that the project was completely and slowly developed by me. I also conclude that this project has helped us gain more knowledge about the topic that we are indulged ourselves into " Visual Studio ". I would be glad to enhance and promote this project if given chance and help ourselves and society in the near future.
* The developed application is tested with sample inputs and outputs obtained in according to the requirement. Even though I have tried our level best to make it a dream project. Due to time
* constraints I could not add more facilities to it.
* The main objective of the Project on Event management System is to manage the details of Event, Activity, Organizers, Attendees, Conductors. It manages all the information about Event, Payment, Conductors, Event. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Event, Activity, Payment, Organizers. It tracks all the details about the Organizers, Attendees, Conductors. it.

**CHAPTER 11**

**REFERENCES**

**Geeksforgeeks :**

<https://www.geeksforgeeks.org/inheritance-in-c/> <https://www.geeksforgeeks.org/c-classes-and-objects/> [https://www.geeksforgeeks.org/constructors-c/](https://www.geeksforgeeks.org/c-classes-and-objects/)

# Code with harry :

<https://youtu.be/j8nAHeVKL08>

# Mysirg :

[https://youtu.be/luo9PpGE04Y](https://youtu.be/j8nAHeVKL08)