LINK OF EXPLANATORY VIDEO AND PROJECT VS FILE :

https://drive.google.com/drive/folders/17memTnCT-nkILDz\_kL43z4Gdhj37hmpA?usp=sharing

**CSL 113 – Computer Programming Lab  
  
BSCS 1 A – Fall 2020**

**[CAR PARKING SYSTEM]**



**Presented by: [-Muhammad Ahsan**

**-Aliyan Rehman**

**-Arham Naeem]**

**Submitted to: Miss Saba Imtiaz and Amna Iftikhar**

**DEPARTMENT OF COMPUTER SCIENCE**

**BAHRIA UNIVERSITY, KARACHI CAMPUS**

TABLE OF CONTENTS:

1. **Introduction3**
   1. Project Description 3
   2. Project Scope4
   3. Project Modules4
   4. Features of Project4-5
2. Requirement5
   1. Software Requirement5
3. Analysis5
   1. Existing System5-6
   2. Proposed System6
4. Disadvantages of the Program6-7
5. System Implementation7
   1. Introduction7
   2. Code7-18
6. Snap-shots19-24

# Introduction:

* 1. Project Description:

Car parking systems are known for its integration with technology. Most systems are based on improved models and technological innovations, because of this, parking management systems are suited to be used in various parking lots.

Comparatively, car parking system is very easy for authorities and vehicle owners to use it. It is also very flexible and does not cause any inconvenience to its users. Additionally, you can adjust the settings depending on the number of vehicle traffic.

The Car Management System is mini software coded in C++ to ease the workload of cashier as software is automated in taking entry and exit timing and calculating the bill.

This Car parking system is designed to facilitate user/admin. This car park system saves time, money, space and simplifies the tedious task of parking. Provides, optimum solutions and smart parking system for you. Lastly, this parking system satisfies parking request at any place. Do you need to park? Where do you need to park? do you need a pay and display system? do you need to de-park? Whatever your need is, an effective parking management system will be able to accommodate this.

* 1. Project Scope:

This program was successfully completed and ran on our operating system. With the growing need for parking, it is important to upgrade to a well-planned parking space. This program aims to make parking task safe, fast, and efficient for the user so if there are proper parking places then traffic problem would be less. We have seen an evolution of technology over the years and part of it is the improvement of parking management systems. Parking Management Systems are not only convenient but flexible when it comes to controlling the flow of vehicles in a parking area. This is a mini software which can run on low specification system is great for mini parking plazas where high-end systems can’t be installed

* 1. Project Modules:
* Function to calculate the fee for parking car.
* Function to open file holding the record of cars that were parked.
* Function to delete the record in file.
* Function to change the password of admin.

## Features of Project:

* In this car parking system, to start the process first there would a choice, if you’re the admin then press 1 to proceed, else 2 if you’re a user.
* After pressing 1, there would a login page open, after entering correct password, options will be displayed for admin.
* If username and password entered doesn’t match or is incorrect, program will display a warning message.
* If 2 is pressed, the process for parking the vehicle will begin for the user, 1 to be pressed for parking and 2 to de-park
* If 1 is pressed then this program will ask about vehicle details, displays the entry time and gives user an ID.
* If 2 is pressed then process for de-parking will begin. First parking ID will be asked and a confirmatory question.
* At the end, program will display the vehicle details that’s been entered and after rounding the amount off, the total amount that is to be paid.

1. Requirements:
   1. Software Requirement:

* VS studio (any version)/code
* Dev C++

1. Analysis:
   1. Existing System:

This car parking system is designed to ease both the user and admin and any kind of car or bike can easily get parked. This program enables user to park his vehicle conveniently by just pressing keys and do not have to wander around the area in search of parking slot. This program generates an ID to user in the beginning, when vehicle is parked, which will be asked when vehicle is needed to be de-parked. This program asks vehicle’s registration no., type and displays the entry time. This program calculates the amount(money) with respect to time, for car its 3RPS/min and for 2RPS/min for bike, if parked for less than 3 hours but if vehicle remains parked for more than 3 hours amount gets doubled. When vehicle is de-parked ID will be asked and data entered of the vehicle is displayed along with the current time the total amount will be calculated by multiplying amount with difference of time. The admin has to log-in through correct password and can check log files it lets admin open the log file check the details of all the cars that’s been de-parked, allows admin to clear the log file, admin can check current rates and can update them and can change its password as well. This program makes the tiring task of parking so convenient.

* 1. Proposed System:

We have tried our best to achieve the targets which we have proposed, the description we proposed is that, In this software we will write such algorithm which will take only car registration number from user where else entry and exit timings and bill will be calculated by program itself and all data will be stored in list and will be erased from list as soon the car leaves. We will do effective data processing requires low system specification and it is cost effective from other software in market.

### Disadvantages of the Program:

* It has a limited car parking slots, after a certain limit no more vehicle can be parked due to lack of space.
* The password is not masked so that it won’t be secure for the admin to write it down in front of anyone else as anyone can see the password
* Any changes in the code written for implementation, would frustrate the program and it won’t be able to run until the error is removed.
* Data is unsecure as this program doesn’t provide any security for the data that user enters.

##### 5.System Implementation:

5.1 Introduction:

The Car Management System is mini software coded in C++ to ease the workload of cashier

5.2 Code:

//Car Parking System CP PROJECT

#include <iostream> //Header files

#include<cstdio>

#include <ctime>

#include <string>

#include <fstream>

#include<cmath>

using namespace std;

//user defined functions

float fee(string, float); //to calculate amount

int out(); //for time is being parked

int in(); //to check if file exist of not

void file(); //to remove/delete the file

void r\_file(); //to change password of admin

string password(); //to change password of admin

struct vehicle\_details //structure variables to hold vehicle details

{

int prkingid; //to hold parking id

int in\_h, in\_m, in\_s; //to hold time in hr,min and sec

string reg\_num; //to hold vehicle registration no.

string veh\_type; //to hold vehicle type car/bike

time\_t hourin; //to hold entry time, when car was parked

time\_t hourout; //to hold time when car is being deparked

float el\_time; //to hold the difference of hourin and hourout

float charges; //to hold amount for parking

};

vehicle\_details veh[12]; //Array to hold parking id

string pass1 = "admin"; //password of the admin

float mnc = 3, mxc = 6, mnb = 2, mxb = 4; //intial prices, globally defined

int main()

{

B:

system("cls");

int USADMIN;

//Initial selection for User/Admin

cout << "============================\n WELCOME TO CAR PARKING \n============================\n";

cout << "1.Admin\n2.User\n3.Exit\nYour Selection...:" << endl;

cin >> USADMIN;

if (USADMIN == 1)

{

C:

system("cls");

string pass;

cout << "Enter password" << endl;

cin >> pass; //Login form for admin

if (pass == pass1) //checks if pass is correct then proceed

{

D:

system("cls"); //clears screen

//Admin screen

cout << "============================\n WELCOME TO CAR PARKING \n============================\n";

int aprs;

//menu

cout << "1.if you want to check log files\n";

cout << "2.if you want to change rates\n";

cout << "3.if you want to see Abouts \n";

cout << "4.Password Change\n5.Goto User End\n6.Exit\nYour Selection...:\n";

cin >> aprs;

if (aprs == 1) //Selection for admin

{

system("cls");

int que;

cout << "1.View Log\n2.Clear Log\n3.Return\nYour Selection...:" << endl;

cin >> que;

if (que == 1) //Open log file contains details that's been parked

{

file();

system("pause");

goto D;

}

else if (que == 2) //Deletes all the details

{

r\_file();

system("pause");

system("cls");

goto D;

}

else if (que == 3) //Jumps back to starting of admin screen

{

goto D;

}

}

else if (aprs == 2) //Enables admin to update the rates

{

system("cls");

cout << "Enter New Minimum Rate For Car" << endl;

cin >> mnc;

cout << "Enter New Maximum Rate For Car" << endl;

cin >> mxc;

cout << "Enter New Minimum Rate For Bike" << endl;

cin >> mnb;

cout << "Enter New Maximum Rate For Bike" << endl;

cin >> mxb;

cout << "Rates Successfully Changed!!!" << endl;

system("pause");

system("cls");

goto D;

}

else if (aprs == 3) // Short detail about this car parking system

{

system("cls");

cout << "\t\t\t\t\t===========================\n\t\t\t\t\t|||||||||||ABOUT|||||||||||\n\t\t\t\t\t===========================\n" <<

" This Car parking system is designed to facilitate user/admin." <<

"This car park system saves time, money, space and\n" << " simplifies the " <<

"tedious task of parking. Provides, optimum solutions and smart parking system for you.\n" <<

" Lastly,this parking system satisfies your parking request at any place.Thank you for having us!\n\n" <<

" Developed by:Ahsan Arshad, Arham Naeem & Aliyan Rehman\n\n";

system("pause");

system("cls");

goto D;

}

else if (aprs == 4) //Enables admin to change password

{

cout << password(); //calls function

system("pause");

goto B;

}

else if (aprs == 5) //Jumps to user screen

{

goto A;

}

else if (aprs == 6) //Exits the program

{

exit(0);

}

else //Warning message that the number pressed is not valid

{

cout << "Invalid Selection!!!" << endl;

system("pause");

goto C;

}

}

//Warning message that the password is not valid

else {

cout << "wrong password" << endl;

goto B; //jumps back to the beginning of the program

}

}

//Admin view ends here

else if (USADMIN == 2) //Displays user's screen

{

A: //user's view

int uprs;

system("cls");

//Selection for user

cout << "============================\n WELCOME TO CAR PARKING \n============================\n";

cout << "1.if you want to Park\n2.if you want to Depark\n3.Goto Admin-end\n4.About\n5.Exit\nYour Selection...:\n"; //menu

cin >> uprs;

if (uprs == 1) //Enables user to park his vehicle

{

//Program for parking is proceeding

int menu;

int vehicle;

int vehicle2;

int x;

time\_t now = time(0); //Displays current time

tm\* time1 = localtime(&now);

cout << "Today's time:" << (time1->tm\_hour) << ":" << (time1->tm\_min) << ":" << (time1->tm\_sec) << endl;

for (x = 1; x <= 13; x++)

{

if (x > 12) //Bound to max limit of 12 vehicles parking slots

{

//Displays message is no slot is free

cout << "No Space Avaliable" << endl;

system("pause");

}

else if (veh[x].prkingid == x) //Checks whether parking id is already given

{

continue;

}

else

{

time\_t now = time(0);

tm\* time1 = localtime(&now); //Displays current time

veh[x].prkingid = x;

cout << "Enter Car Registration (AAA-0000)" << endl;

cin >> veh[x].reg\_num; //Takes input, registration number of the car

E:

//Selection for user

cout << "Enter Vehicle Type\n1.Car\n2.Bike\n\nYour Selection...:" << endl;

cin >> vehicle;

if (vehicle == 1)//if 1 is entered vehicle is car

{

veh[x].veh\_type = "car";

}

else if (vehicle == 2)//If you enter 2 the vehicle will be bike

{

veh[x].veh\_type = "bike";

}

//if wrong number is pressed

else { cout << "Invalid Selection Re check !! " << endl; goto E; //jumps back to selection

}

//entry time being stored in a variable

veh[x].in\_h = (time1->tm\_hour); veh[x].in\_m = (time1->tm\_min); veh[x].in\_s = (time1->tm\_sec);

cout << "Entry Time...:" << veh[x].in\_h << ":" << veh[x].in\_m << ":" << veh[x].in\_s << endl;

cout << "YOUR PARKING ID IS " << x << "" << endl; //Displays parking ID

veh[x].hourin = in();

system("pause"); //Pauses the program

}

goto A; // Jumps back to beginning of user's screen

}

}

else if (uprs == 2) //if 2 is pressed

{

//Program for deparking is proceeding

int prking;

char quest;

string type;

float tot\_time;

cout << "Enter Parking Id....:" << endl; //enter the correct parking id to get your car deparked

cin >> prking;

if (prking == veh[prking].prkingid)//if parking id is correct or exist

{

//Confirmatory question

cout << "Do you really want to depark?(y/n)" << endl;

cin >> quest;

if (quest == 'y') //If want to depark

{

time\_t now1 = time(0); //calculates time

tm\* time2 = localtime(&now1);

ofstream fs("D://logs.txt", ios::out | ios::app); //Writes information onto the file

cout << "Parking Id...:" << veh[prking].prkingid << endl; //displays the parking id which program has generated while vehicle was being parked

cout << "Registration Number...:" << veh[prking].reg\_num << endl; //displays the reg no

cout << "Vehicle Type...:" << veh[prking].veh\_type << endl; //displays the vehicle type car/bike

//displays the time when car was parked

cout << "Entry Time...:" << veh[prking].in\_h << ":" << veh[prking].in\_m << ":" << veh[prking].in\_s << endl;

veh[prking].hourout = out();

//displays the current time when car is being deparked

cout << "Exit Time...:" << (time2->tm\_hour) << ":" << (time2->tm\_min) << ":" << (time2->tm\_sec) << endl;

//difftime calculates the difference of time when car is being deparked

//and when car was parked in seconds to calculate the amount

veh[prking].el\_time = difftime(veh[prking].hourout, veh[prking].hourin) / 60; //"/60" converts the time to minute as function returns time in seconds

cout << "Total Time(In Minutes)" << veh[prking].el\_time << endl; //displays the time after differentitation

type = veh[prking].veh\_type;

tot\_time = veh[prking].el\_time;

veh[prking].charges = fee(type, tot\_time);

cout << "Total Charges Rs." << roundf(veh[prking].charges) << endl; //displays total charge

//Data of the vehicle is being entered in file

fs << endl;

fs << "Parking Id...:" << veh[prking].prkingid << endl;

fs << "Registration Number...:" << veh[prking].reg\_num << endl;

fs << "Vehicle Type...:" << veh[prking].veh\_type << endl;

fs << "Entry Time...:" << veh[prking].in\_h << ":" << veh[prking].in\_m << ":" << veh[prking].in\_s << endl;

fs << "Exit Time...:" << (time2->tm\_hour) << ":" << (time2->tm\_min) << ":" << (time2->tm\_sec) << endl;

fs << "Total Time(In Minutes)" << veh[prking].el\_time << endl;

fs << "Total Charges Rs." << roundf(veh[prking].charges) << endl; //rounds off the price and returns

fs << endl;

fs.close(); //closes the file

veh[prking].prkingid = 0;

veh[prking].reg\_num = "";

veh[prking].veh\_type = "";

veh[prking].hourin = 0;

veh[prking].hourout = 0;

veh[prking].el\_time = 0;

veh[prking].charges = 0;

system("pause");

goto A;

}

else if (quest == 'n') //if no need to depark, then jumps back to A

{

goto A;

}

}

else

{

cout << "Invalid Id!!!" << endl; //If wrong id is entered display a warning message

system("pause");

goto A;

}

}

else if (uprs == 3) //if 3 is pressed jump to admin screen

{

system("cls");

goto C;

}

else if (uprs == 4) //displays short detail about this program

{

system("cls");

cout << "\t\t\t\t\t===========================\n\t\t\t\t\t|||||||||||ABOUT|||||||||||\n\t\t\t\t\t===========================\n" <<

" This Car parking system is designed to facilitate user/admin." <<

"This car park system saves time, money, space and\n" << " simplifies the " <<

"tedious task of parking. Provides, optimum solutions and smart parking system for you.\n" <<

" Lastly,this parking system satisfies your parking request at any place.Thank you for having us!\n\n" <<

" Developed by:Ahsan Arshad, Arham Naeem & Aliyan Rehman\n\n";

system("pause");

system("cls");

goto A;

}

else if (uprs == 5) //exits the program

{

exit(0);

}

else //if wrong number is pressed, warning message

{

cout << "Invalid Selection!!!" << endl;

system("pause");

goto A;

}

}

//User view ends

//Choice for user/admin

else if (USADMIN == 3)

{

exit(0); //exits the program

}

else //warning message

{

cout << "Invalid Id!!!" << endl;

system("pause");

system("cls");

goto B; //jumps to beginning of the program

}

}

//FUNCATIONS BEING USED

float fee(string type, float tot\_time) //calulates the amount(total fee)

{

float charges, minc, maxc, minb, maxb;

minc = mnc; // minimum price for car

maxc = mxc; // maximum price for car

minb = mnb; // minimum price for bike

maxb = mxb; // maximum price for bike

if (type == "car")

{

if (tot\_time < 180) //price,if car is parked for less than 3 hours

{

charges = (tot\_time \* minc); //multiplies difference of time with minc

return charges;

}

else if (tot\_time > 180) //if car is parked for more than 3 hours, doubles the price

{

charges = (tot\_time \* maxc); //multiplies difference of time with maxc

return charges;

}

}

else if (type == "bike")

{

if (tot\_time < 180) //price if bike is parked for less than 3 hours

{

charges = (tot\_time \* minb); //multiplies difference of time with minb

return charges;

}

else if (tot\_time > 180)//if bike is parked for more than 3 hours, doubles the price

{

charges = (tot\_time \* maxb); //multiplies difference of time with maxb

return charges;

return charges;

}

}

}

int out() //displays time of when car is getting deparked

{

time\_t hourout;

return time(&hourout);

}

int in() //displays time of when car got parked

{

time\_t hourin;

return time(&hourin);

}

void file() //checks if file exist or not

{

fstream files;

files.open("D://logs.txt"); //opens file in D drive

files.close();

if (files.fail())

{

cout << "Log file doesn't exists" << endl;

}

else

{

system("notepad.exe D://logs.txt");

}

}

void r\_file()//Removes/deletes the file and all the data in it

{

fstream files;

files.open("D://logs.txt");

files.close();

if (files.fail())

{

cout << "Log file doesn't exists" << endl;

}

else

{

remove("D://logs.txt");

cout << "Log Cleared!!" << endl;

}

}

string password() //changes password

{

cout << "New Password..>:" << endl;

cin >> pass1;

return "Password Changed"; //password changes

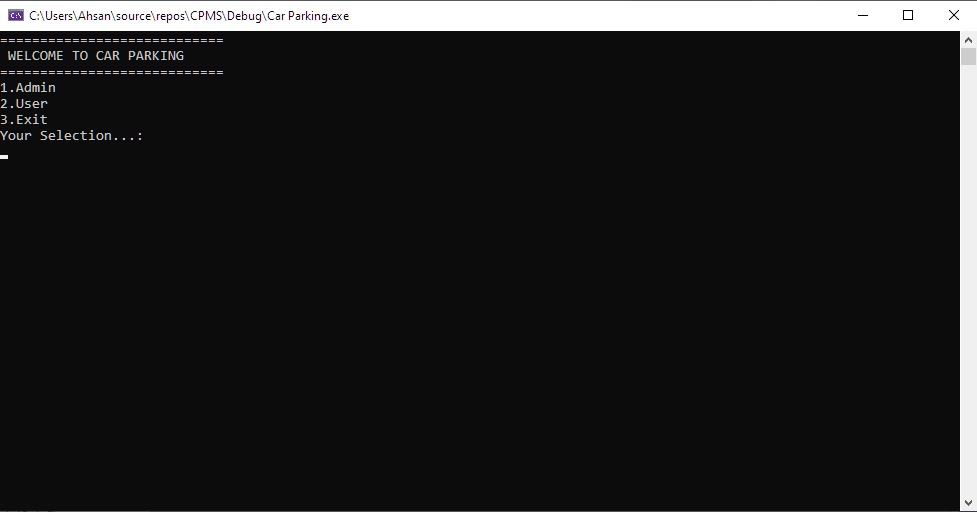
}

//END OF THE PROGRAM

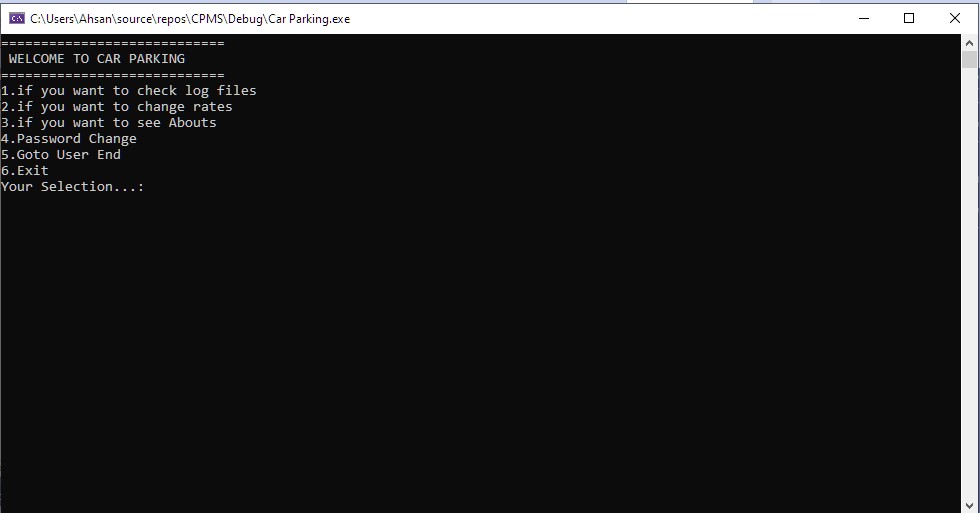
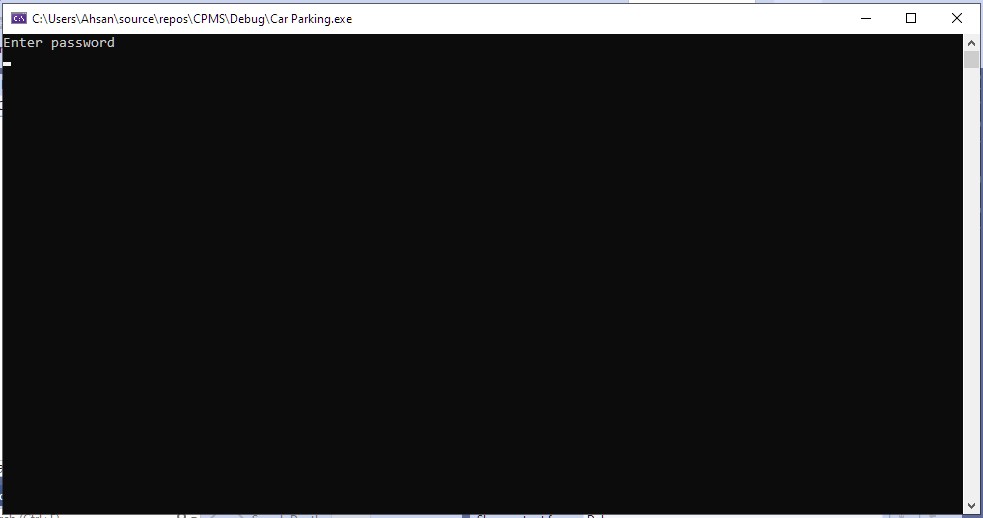
//Code commented by Arham Naeem

//Report done by Arham Naeem and Aliyan Rehman

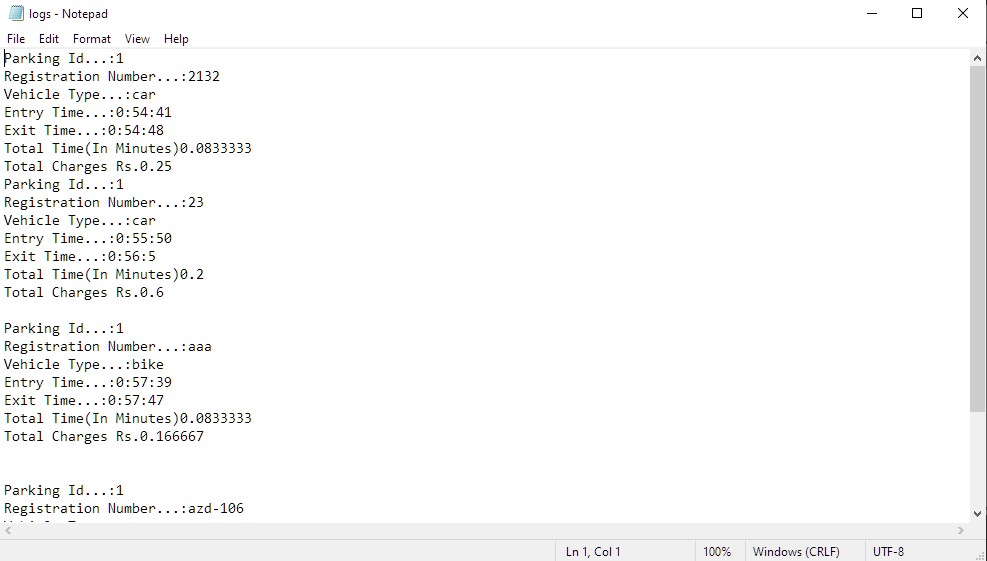
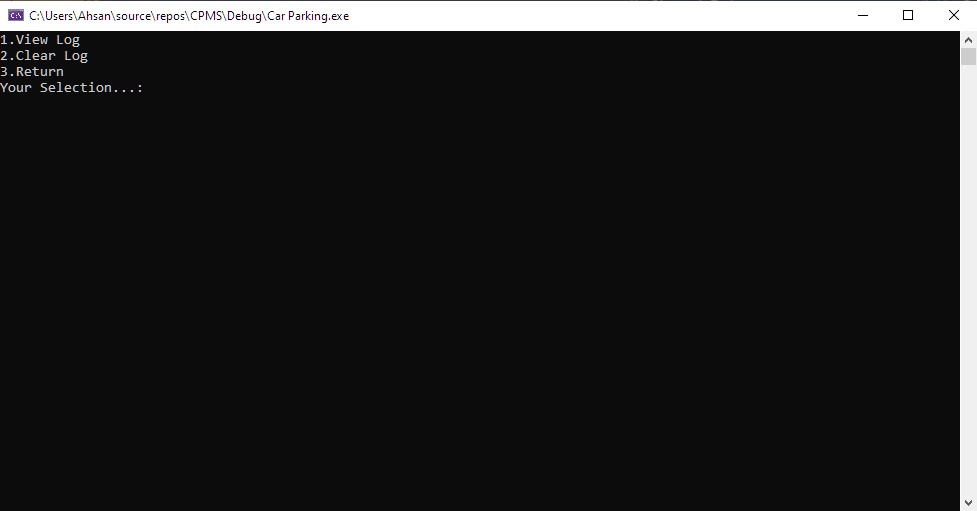
###### 6.SNAP-SHOTS(PROGRAM OUTPUT):

1. Home Page

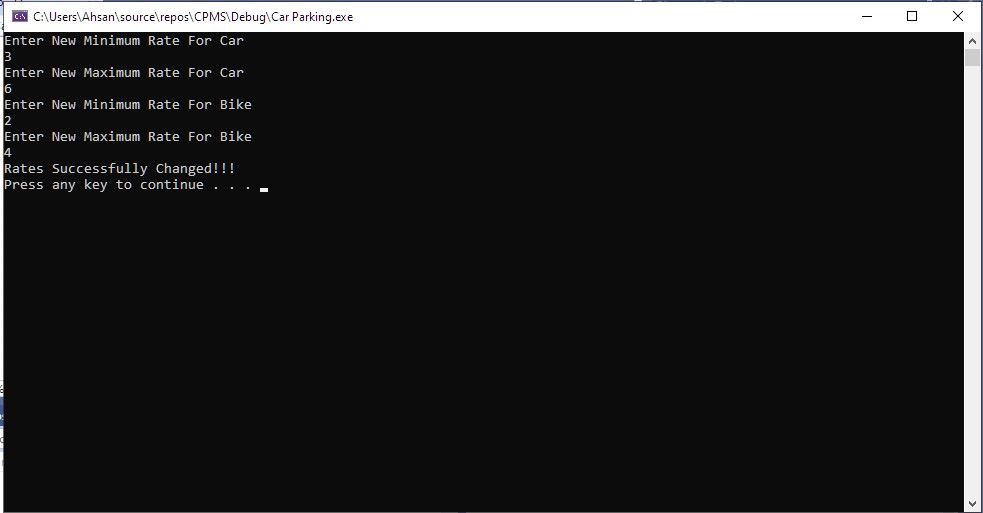
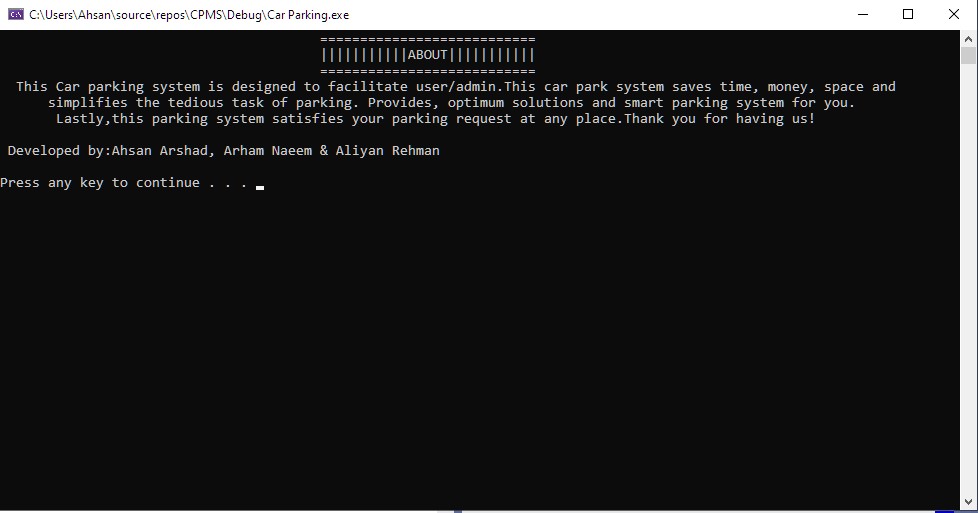
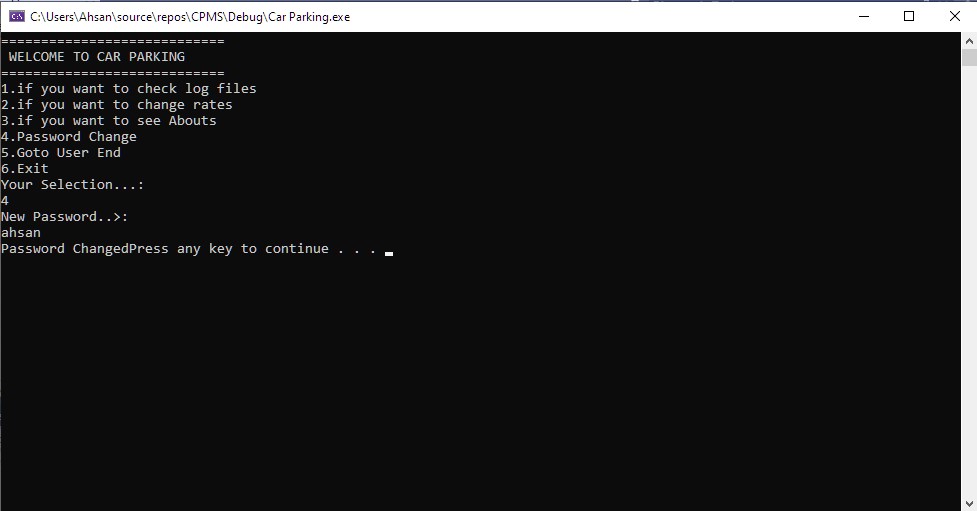
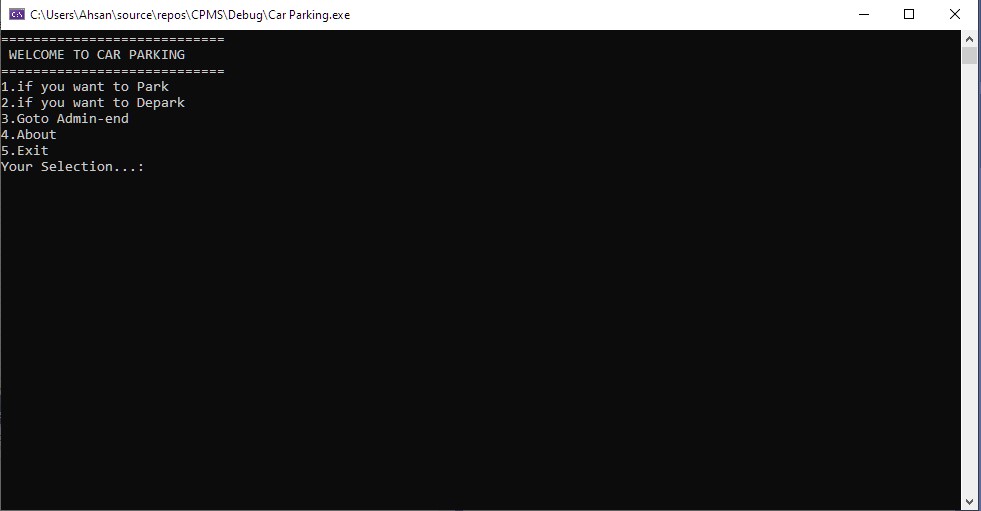
The First Screen After Running of program.

1. Admin End

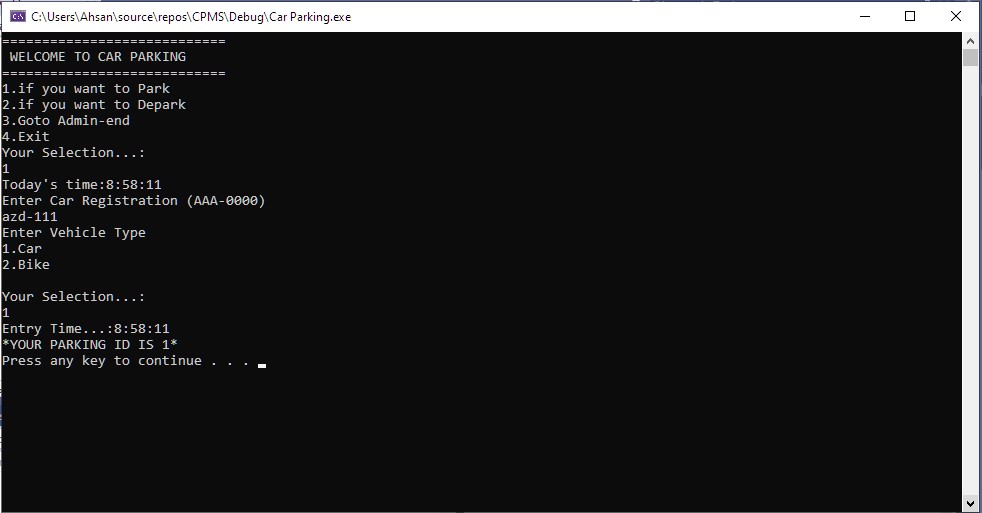
Admin-end is secured with password. Gives admin access to permanent log created of parked vehicles abilities to change rates, to change password.

1. Log Files Tab

The log created of parked vehicle.

1. Tariff Change
2. Abouts
3. Password Change
4. User End 

Specified for user gives it access to enter parking lot with just inserting minor car details. and exiting with just adding parking id generated at time of entry.

1. Parking 
2. De-parking