

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY NOIDA, Sector-128

B. TECH CSE - SEMESTER 2

SDF LAB-2

SUBJECT CODE: 15B17CI271



(Project Report)

“CAR RENTAL SYSTEM using C++”

Submitted By:

Anand Panda : 9922103099

Ansh Bansal : 9922103097

Rahul Yadav : 9922103091

(Batch – F4)

Submitted To:

Dr.Shailesh Kumar

Mr.Shariq Murtuza

INDEX

S.No	Topic	Page No
1.	Title of the Project	2
2.	Abstract of the Project	3-4
3.	Topics of SDF-1 used	5
4.	Flowchart	6
5.	Functions and library used	7-8
6.	Output of program	9-12
7.	Source code	12-27
8.	Conclusion	28

CAR RENTAL SYSTEM

CAR RENTAL SYSTEM

Introduction:

Car Rental System

This is a project for Car Rental System in C++. A user can perform many kinds of management tasks to manage rent and car records like adding a new car as well as modifying and removing it. Also, users can view all rental records, check car details as well as rent and change rent details from rental records. The system automates the basic car rental functions to aid in the day-to-day operations of a car rental staff. Also, the goal of the system is to allow for storing details of a large number of car details and allow for add, retrieve and update details of any car.

- **Add Car:** This operation is necessary when user adds a new car. This option takes inputs of car details.
- **Update Car Details:** This function updates details of car in the car records.
- **Remove Car:** This operation clears the existing record of a car from the records. So, be careful while performing this operation because the system could lose any important data.
- **List all cars:** Show all the available cars from the record of car rental system.
- **Check Car**
- **Rent a Car**
- **Modify Rent Records**

This system is a small project that implements only some of the complex features of car rental service. However, one can use this project to make complex system by adding more details/properties to the records as well as features and functions to the system.

Objective:

- To simplify the task of maintaining huge chunks of records of cars.
- To develop a database to store cars renting information.
- To provide functional report to management of car rental agency

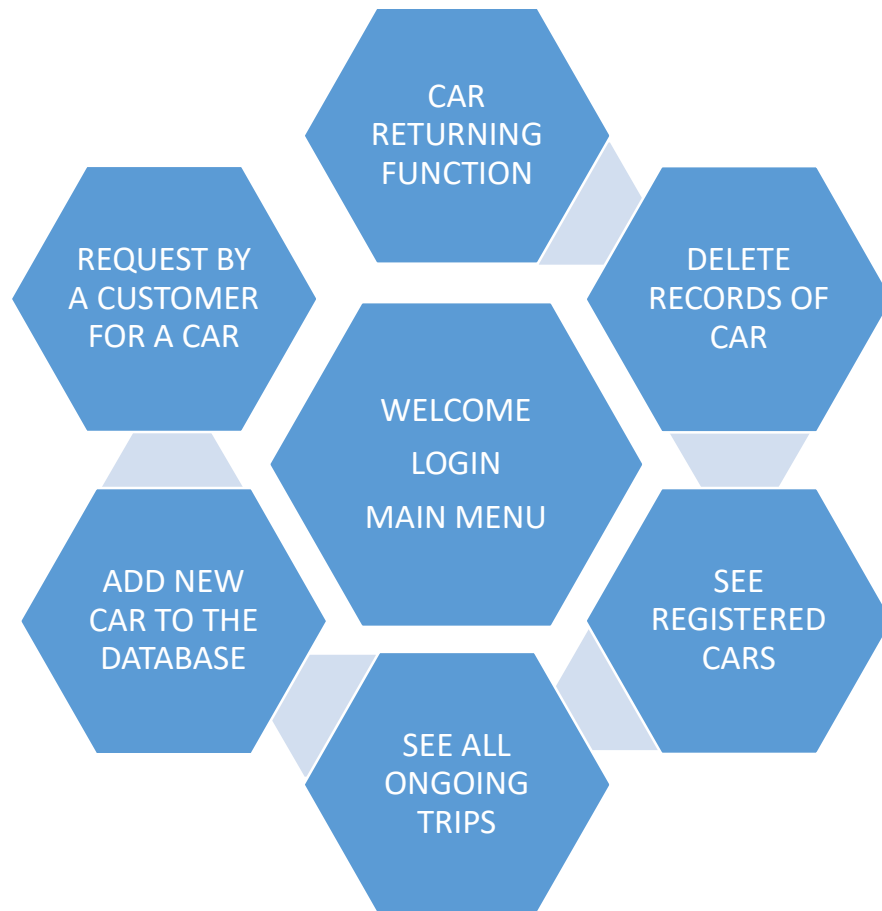
Features:

- LOGIN SCREEN
- PASSWORD
- CAR RETURNING FUNCTION
- ADD NEW CAR TO THE DATABASE
- DELETE RECORDS OF CAR
- SEE ALL ONGOING TRIPS
- SEE REGISTERED CARS

Topics of SDF used:

1. Header Files
2. Data Types
3. Exception handling
4. If-else statements
5. Do While loop/For loop/while loop
6. Switch case
7. Functions
8. Get function
9. File Handling
10. Manipulating buffer input and output stream
11. Getch function to push buffer window
12. Classes and objects concept
13. Friend function
14. Templates
15. Member function
16. OOPS properties --- >
17. Inheritance
18. Data hiding
19. Polymorphism

FLOWCHART



FUNCTIONS AND LIBRARY USED

- **#include <iostream>** : Library of c++
- **#include <fstream>** : This class is the combination of both ofstream and ifstream. It provides the capability of creating, writing and reading a file.
- **#include <stdlib.h>** : Declares functions for memory functions and algorithms
- **#include <conio.h>** : It is used to include the console input output library functions.
- **#include <iomanip>** : It is used to set basefield flag. It is used to set fill character. It is used to set decimal precision.
- **#include <cstring>** : This header file defines several functions to manipulate C strings and arrays
- **cin.getline()** : To take input from user and to avoid null exception of strings
- **fflush(stdin)** : To clear the input stream
- **template <class T>** : Declaring template class
- **try** : Encapsule the code that may raise error
- **throw** : Throw error when it is caught during run time
- **catch** : Catches error and handle it and also transfers the natural flow of program
- **setw()** : To set the width to be printed on console using cout
- **cout << left** : This is used to add extra spaces after text corresponding to the setw width
- **input.eof** : To check if the file reaches to end

- **input.open** : To open file and declare a object pointer to it
- **input.close** : To close the file
- **strcmp** : To compare two strings
- **fflush(stdout)** : To clear the output buffer stream

OUTPUT OF FUNCTIONS

WELCOME SCREEN

```
                                WELCOME

                                -----
                                | ADMIN LOGIN |
                                -----
                                Password : *****
                                Try Again

Press any key to continue . . . █
```

MAIN MENU

```
                CAR RENTAL MANAGEMENT SYSTEM.
                -----
What would you like to do Sir?
1. Register new car to the Database.
2. See all registered cars.
3. Customer request for a car.
4. Customer wants to return a car.
5. See all ongoing trips.
6. Delete a car from database.
7. LogOut.

Enter your choice :
```

SEE ALL ONGOING TRIPS

```
                Showing all ongoing trips.

                -----
|Customer name       : ANSH BANSAL |
|Customer contact number : 9953249705 |
|Rented car plate number : 1234 |
|Time period        : 2 |
|Charges            : 60 |
|Total charges(including tax): 132 |
                -----
| NO FURTHERE RECORDS |
                -----

Total cars : 1
Press any key to continue . . .
```

SEE ALL REGISTERED CARS

```
Showing all available Cars.

-----
|Enter company name           : maruti suzuki|
|Enter model name             : waganor      |
|Enter plate number           : 1234         |
|Capacity (5 or 7 seater)     : 5           |
|Fuel(P petrol, D diesel, C CNG) : p       |
|Color                         : grey        |
|-----|
|                               NO FURTHURE RECORDS                               |
|-----|
Total cars : 1
Press any key to continue . . .
```

CUSTOMER REQUEST FOR A CAR

```
Showing all available Cars.

-----
|Enter company name           : maruti suzuki|
|Enter model name             : waganor      |
|Enter plate number           : 1234         |
|Capacity (5 or 7 seater)     : 5           |
|Fuel(P petrol, D diesel, C CNG) : p       |
|Color                         : grey        |
|-----|
|                               NO FURTHURE RECORDS                               |
|-----|
Total cars : 1
Press any key to continue . . .
Customer is requesting for a car.
Enter Customer Details.
Customer Name       : ANSH BANSAL
Contact Number      : 9953249705
Choose plate number : 1234
Car of plate number 1234 is available.
Enter Time period(in hours or in days) : 2
Enter charges(per hours or per day)     : 60
Total charges(including 10%Tax)         : 132

Trip has been started.
List of available cars has been updated.
Have a new customer (y or n) :
```

CUSTOMER WANTS TO RETURN A CAR

```
Showing all ongoing trips.

-----
|Customer name           : ANSH BANSAL           |
|Customer contact number : 9953249705             |
|Rented car plate number  : 1234                 |
|Time period             : 2                     |
|Charges                  : 60                    |
|Total charges(including tax): 132                |
-----
|                                NO FURTHERE RECORDS                                |
-----

Total cars : 1
Press any key to continue . . .

Enter plate number of returned car : 1234
Showing details of trips.

-----
|Customer name           : ANSH BANSAL           |
|Customer contact number : 9953249705             |
|Rented car plate number  : 1234                 |
|Time period             : 2                     |
|Charges                  : 60                    |
|Total charges(including tax): 132                |
-----

List of available has been updated.
Ongoing trips has been updated.
Is their another trip to end(y/n) :
```

REGISTER A NEW CAR TO THE DATABASE

```
Registering new cars to our Database.
-----

Would you like to delete all previous records(y or n) ?y
In in new file
Enter company name           : xyz
Enter model name             : creta
Enter plate number           : 5647
Capacity (5 or 7 seater)     : 5
Fuel(P petrol, D diesel, C CNG) : c
Color                        : blue
Car added to database successfully.
Would you like to enter new car to database(y or n): n
Press any key to continue . . .
```

DELETE A CAR

```
Showing all available Cars.

-----
|Enter company name      : xyz      |
|Enter model name       : creta    |
|Enter plate number     : 5647     |
|Capacity (5 or 7 seater) : 5      |
|Fuel(P petrol, D diesel, C CNG) : c |
|Color                  : blue     |
|-----|
|                          NO FURTHURE RECORDS                          |
|-----|
Total cars : 1
Press any key to continue . . .

Enter plate number : 5647
Press any key to continue . . .
```

SOURCE CODE

```
#include <iostream>
#include <fstream>
#include <stdlib.h>
#include <conio.h>
#include <iomanip>
#include <cstring>

using namespace std;
void adminLogin();
void adminPage();
void addCar();
class Cars
{
    public:
        char plate_no[15];    // eg. UP15AA0000 & we right code to make it
unique
        char company_name[20];
        char model_name[20];
        char fuel_type;      // have only three option :- P for petrol, D for
desile, C for CNG
        char seater;        // have only two option 5, 7 seater
        char color[15];
        public:
        void entry()
        {
            cout << "Enter company name          : "; fflush(stdin);
cin.getline(company_name, 18);
            cout << "Enter model name            : "; fflush(stdin);
cin.getline(model_name, 18);
            cout << "Enter plate number          : "; fflush(stdin);
cin.getline(plate_no, 12);
            cout << "Capacity (5 or 7 seater)      : "; fflush(stdin); cin >> seater;
            cout << "Fuel(P petrol, D diesel, C CNG) : "; fflush(stdin); cin >>
fuel_type;
            cout << "Color                        : "; fflush(stdin); cin.getline(color,
14);

        }
        friend bool check(char plate_no[]);
        friend int displaydata();
};
```

```

template <class T>
class customer
{
    public:
    char name[25];
    char phone_no[15];
    char plate_no[15];
    T time_per;
    T charges;
    void getData()
    {
        cout << "Enter Customer Details." << endl;
        cout << "Customer Name      : "; fflush(stdin); cin.getline(name,
20);
        cout << "Contact Number    : "; fflush(stdin); cin.getline(phone_no,
12);
    }
    friend void requestcar();

};

int displaydata(char opt)
{
    Cars show; int counting = 0;
    try
    {
        ifstream input;
        if(opt == '*')
        {
            input.open("carsDatabas.dat", ios :: binary | ios :: in);
        }
        else
        {
            input.open("carsAvailable.dat", ios :: binary | ios :: in);
        }
        if(!input.is_open())
        {
            throw 1;
        }
        input.seekg(ios :: beg);
        system("CLS");
        cout << "\t Showing all available Cars." << endl << endl;
        cout << " ----- " <<
endl;

```

```

while(input.read((char *)&show, sizeof(show)))
{
    counting++;

    cout << " |Enter company name          : "<< setw(30) << left <<
show.company_name << "|" << endl;
    cout << " |Enter model name          : "<< setw(30) << left <<
show.model_name << "|" << endl;
    cout << " |Enter plate number        : "<< setw(30) << left <<
show.plate_no << "|" << endl;
    cout << " |Capacity (5 or 7 seater)    : "<< setw(30) << left <<
show.seater << "|" << endl;
    cout << " |Fuel(P petrol, D diesel, C CNG) : "<< setw(30) << left
<< show.fuel_type << "|" << endl;
    cout << " |Color                      : "<< setw(30) << left <<
show.color << "|" << endl;
    cout << " -----"
<< endl;
}
if(input.eof() && counting != 0)
{
    cout << " |                NO FURTHERE RECORDS
|" << endl;
    cout << " -----"
<< endl;
}
else if( input.eof() && counting == 0)
{
    cout << " |                ADD NEW CARS TO DATABASE
|" << endl;
    cout << " -----"
<< endl;
}
cout << "Total cars : " << counting << endl;
input.close();

}
catch(...)
{
    cout << "Some Technical Error occurred, contact Developer ASAP."
<< endl;
    system("PAUSE");
}

```



```

    }
    system("PAUSE");
    return counting;
}

bool check(char plate_no[], char mode)
{
    Cars chek;
    try
    {
        ifstream input;
        if(mode == '*')
        {
            input.open("carsDatabas.dat", ios :: binary | ios :: in );
        }
        else
        {
            input.open("carsAvailable.dat", ios :: binary | ios :: in);
        }
        if(!input.is_open())
        {
            throw 1;
        }
        input.seekg(ios :: beg);
        while(!input.eof())
        {
            input.read((char *)&chek, sizeof(chek));

            if(strcmp(plate_no, chek.plate_no) == 0)
            {
                input.close();
                return 0;
            }
        }
        input.close();
        return 1;
    }
    catch(...)
    {
        cout << "Some Technical Error occurred, contact Developer ASAP."
        << endl;
        system("PAUSE");
    }
}

```

```

    return 0;
}
void update(char plate_no[])
{
    Cars chek;
    try
    {
        ifstream input; ofstream temp;
        input.open("carsAvailable.dat", ios :: binary | ios :: in);
        temp.open("temporary.dat", ios :: binary | ios :: out | ios :: trunc);
        if(!input.is_open())
        {
            throw 1;
        }
        input.seekg(ios :: beg); //temp.seekp(ios :: end);
        while(input.read((char *)&chek, sizeof(chek)))
        {
            //input.read((char *)&chek, sizeof(chek));

            if(strcmp(plate_no, chek.plate_no) != 0)
            {
                temp.write((char *)&chek, sizeof(chek));
            }
        }
        input.close();
        remove("carsAvailable.dat");
        temp.close();

        //newrec();
        ofstream out; ifstream in; Cars copyy;
        out.open("carsAvailable.dat", ios :: binary | ios :: out | ios :: trunc);
        //cout << "Checkpoint 1" << endl;
        in.open("temporary.dat", ios :: binary | ios :: in);
        in.seekg(ios :: beg); //out.seekp(ios :: end);
        while(in.read((char *)&copyy, sizeof(copyy)))
        {
            out.write((char *)&copyy, sizeof(copyy));
        }
        out.close();
        in.close();
    }
    catch(...)

```

```

        {
            cout << "Some Technical Error occurred, contact Developer ASAP."
<< endl;
            system("PAUSE");
        }
    }
void begintrip(customer<float> confirm)
{
    ofstream out;
    cout << endl;
    out.open("ongoingTrips.dat", ios :: binary | ios :: out | ios :: app);
    out.seekp(ios :: beg);
    try
    {
        if(!out.is_open())
        {
            throw 1;
        }
        out.write((char *)&confirm, sizeof(confirm));
        update(confirm.plate_no);
        cout << "Trip has been started." << endl;
        cout << "List of available cars has been updated." << endl;
    }
    catch(...)
    {
        cout << "Some technical error occurred, contact developer ASAP."
<< endl;
    }
    out.close();
}
void requestcar()
{
    Cars req; int empty = 0; customer<float> cust; char choice = 'y';
    starting:
    fflush(stdout);
    //cout << "Customer is requesting for car." << endl;

    empty = displaydata('0');
    cout << "Customer is requesting for a car." << endl;
    if(empty != 0)
    {
        cust.getData();
    }
}

```

```

        cout << "Choose plate number : "; fflush(stdin);
cin.getline(cust.plate_no, 12);
    if(!check(cust.plate_no, 'a'))
    {
        cout << "Car of plate number " << cust.plate_no << " is
available." << endl;
        cout << "Enter Time period(in hours or in days) : "; fflush(stdin);
cin >> cust.time_per;
        cout << "Enter charges(per hours or per day)  : "; fflush(stdin);
cin >> cust.charges;
        cout << "Total charges(including 10%Tax)      : " <<
(cust.time_per * cust.charges) + (0.10 * cust.time_per * cust.charges) <<
endl;
        begintrip(cust);
    }
    else
    {
        cout << "Sorry Car of requested plate number is not registered
with us." << endl;
    }
    cout << "Have a new customer (y or n) : "; fflush(stdin);cin >>
choice;
    if(choice == 'y')
    {
        goto starting;
    }
}
else
{
    cout << "Sorry no service available. Come back later." << endl;
}
system("PAUSE");

}
int showtrips()
{
    customer<float> show; int counting = 0;
    try
    {
        ifstream input;
        input.open("ongoingTrips.dat", ios :: binary | ios :: in);
        if(!input.is_open())
        {

```

```

        throw 1;
    }
    input.seekg(ios :: beg);
    system("CLS");
    cout << "\t Showing all ongoing trips." << endl << endl;
    cout << " -----"
<< endl;
    while(input.read((char *)&show, sizeof(show)))
    {
        counting++;

        cout << " |Customer name          : "<< setw(30) << left <<
show.name << "|" << endl;
        cout << " |Customer contact number    : "<< setw(30) << left <<
show.phone_no << "|" << endl;
        cout << " |Rented car plate number      : "<< setw(30) << left <<
show.plate_no << "|" << endl;
        cout << " |Time period                : "<< setw(30) << left <<
show.time_per << "|" << endl;
        cout << " |Charges                    : "<< setw(30) << left <<
show.charges << "|" << endl;
        cout << " |Total charges(including tax): "<< setw(30) << left <<
(show.time_per * show.charges) + (0.10 * show.time_per *
show.charges) << "|" << endl;
        cout << " -----"
<< endl;
    }
    if(input.eof() && counting != 0)
    {
        cout << " |                      NO FURTHERE RECORDS
|" << endl;
        cout << " -----"
<< endl;
    }
    else if( input.eof() && counting == 0)
    {
        cout << " |                      ADD CUSTOMER TO DATABASE
|" << endl;
        cout << " -----"
<< endl;
    }
    cout << "Total cars : " << counting << endl;

```

```

        input.close();

    }
    catch(...)
    {
        cout << "Some Technical Error occurred, contact Developer ASAP."
<< endl;
        system("PAUSE");
    }

    system("PAUSE");
    return counting;
}
void returncar()
{

    int empty; int exist;
    char choice;
    char plate_no[15];
    customer<float> rtn;
    Cars back;
    starting:
    empty = showtrips(); exist = 0; choice = 'y';
    if(empty != 0)
    {
        cout << endl << "Enter plate number of returned car : ";fflush(stdin);
cin.getline(plate_no, 12);
        ifstream input;
        input.open("ongoingTrips.dat", ios :: binary | ios :: in);
        ofstream temp;
        temp.open("temporary.dat", ios :: binary | ios :: out | ios :: trunc);
        while(input.read((char *)&rtn, sizeof(rtn)))
        {
            if(strcmp(plate_no, rtn.plate_no) == 0)
            {
                exist = 1;
                cout << "\tShowing details of trips." << endl << endl;
                cout << " -----
- " << endl;
                cout << " |Customer name          : "<< setw(30)<< left <<
rtn.name << "|" << endl;
                cout << " |Customer contact number    : "<< setw(30)<< left
<< rtn.phone_no << "|" << endl;

```

```

        cout << " |Rented car plate number    : "<< setw(30) << left <<
rtrn.plate_no << "|" << endl;
        cout << " |Time period                : "<< setw(30) << left <<
rtrn.time_per << "|" << endl;
        cout << " |Charges                    : "<< setw(30) << left <<
rtrn.charges << "|" << endl;
        cout << " |Total charges(including tax): "<< setw(30) << left
<< (rtrn.time_per * rtrn.charges) + (0.10 * rtrn.time_per * rtrn.charges)
<< "|" << endl;
        cout << " -----
- " << endl;

    }
    else
    {
        temp.write((char *)&rtrn, sizeof(rtrn));
    }

}
temp.close();
input.close();
if(exist == 1)
{

    input.open("carsDatabas.dat", ios :: binary | ios :: in);
    input.seekg(ios :: beg);
    while(input.read((char *)&back, sizeof(back)))
    {
        if(strcmp(plate_no, back.plate_no) == 0)
        {
            ofstream editt;
            editt.open("carsAvailable.dat", ios :: binary | ios :: out | ios ::
app);

            editt.seekp(ios :: end);
            editt.write((char *)&back, sizeof(back));
            editt.close();
            cout << "List of available has been updated." << endl;
        }
    }
    input.close();
}

```

```

        //Updating ontrip list.
        remove("ongoingTrips.dat");
        ofstream updat;
        updat.open("ongoingTrips.dat", ios :: binary | ios :: out);
        input.open("temporary.dat", ios :: binary | ios :: in);
        input.seekg(ios :: beg);
        while(input.read((char *)&rtrn, sizeof(rtrn)))
        {
            updat.write((char *)&rtrn, sizeof(rtrn));

        }
        cout << "Ongoing trips has been updated." << endl;

    }
    else
    {
        cout << "Car of " << plate_no << " is not gone for trip." << endl;
    }
}
stop2:
cout << "Is their another trip to end(y/n) : ";
fflush(stdin);
cin >> choice;
if(choice != 'y' && choice != 'n')
{
    cout << "\t Invalid input. Try again" << endl;
    goto stop2;
}
else if(choice == 'y')
{
    goto starting;
}
system("PAUSE");
}
void deletecar()
{
    char plate_no[15];
    displaydata('0');
    cout << endl << "Enter plate number : "; fflush(stdin);
    cin.getline(plate_no, 12);
}

```



```

if(!check(plate_no, '0'))
{
    update(plate_no);
    Cars chek;
    try
    {
        ifstream input; ofstream temp;
        input.open("carsDatabas.dat", ios :: binary | ios :: in);
        temp.open("temporary.dat", ios :: binary | ios :: out | ios :: trunc);
        if(!input.is_open())
        {
            throw 1;
        }
        input.seekg(ios :: beg); //temp.seekp(ios :: end);
        while(input.read((char *)&chek, sizeof(chek)))
        {
            //input.read((char *)&chek, sizeof(chek));

            if(strcmp(plate_no, chek.plate_no) != 0)
            {
                temp.write((char *)&chek, sizeof(chek));
            }
        }
        input.close();
        remove("carsDatabas.dat");
        temp.close();

        //newrec();
        ofstream out; ifstream in; Cars copyy;
        out.open("carsDatabas.dat", ios :: binary | ios :: out | ios :: trunc);
        //cout << "Checkpoint 1" << endl;
        in.open("temporary.dat", ios :: binary | ios :: in);
        in.seekg(ios :: beg); //out.seekp(ios :: end);
        while(in.read((char *)&copyy, sizeof(copyy)))
        {
            out.write((char *)&copyy, sizeof(copyy));
        }
        out.close();
        in.close();

    }
    catch(...)
    {

```

```

        cout << "Some Technical Error occurred, contact Developer
ASAP." << endl;
        system("PAUSE");
    }

}
else
{
    cout << "Car of plate number " << plate_no << " is not available.";

}

system("PAUSE");

}
void addCar()
{
    Cars regis; char choice = 'y'; char ans;
    ofstream out; ofstream extra; ofstream extra2;
    system("cls");
    cout << endl;
    cout << "\t Registering new cars to our Database." << endl;
    cout << "\t -----" << endl;
    cout << endl;
    cout << endl;
    cout << "Would you like to delete all previous records(y or n) ?";
    fflush(stdin); cin >> ans;
    if(ans == 'n')
    {
        out.open("carsDatabas.dat", ios :: binary | ios :: out | ios :: app);
        cout << "In append mode." << endl;
        out.seekp(ios :: end);
    }
    else
    {
        out.open("carsDatabas.dat", ios :: binary | ios :: out | ios :: trunc);
        extra.open("carsAvailable.dat", ios :: binary | ios:: out | ios :: trunc);
        extra2.open("ongoingTrips.dat", ios :: binary | ios:: out | ios :: trunc);
        cout << "In in new file" << endl;
    }
    try
    {
        if(!out.is_open())

```

```

        {
            throw 1;
        }
    do
    {
        stop1:
        regis.entry();
        extra.close();
        out.close();
        if(check(regis.plate_no, '*'))
        {
            cout << "\t Car added to database successfully." << endl;
            out.open("carsDatabas.dat", ios :: binary | ios :: out | ios :: app);
            extra.open("carsAvailable.dat", ios :: binary | ios :: out | ios ::
app);
            extra.seekp(ios :: end);
            out.seekp(ios :: end);
            out.write((char *)&regis, sizeof(regis));
            extra.write((char *)&regis, sizeof(regis));
            out.close();
            extra.close();
        }
        else
        {
            cout << "A car with this plate number is already registered.";
        }
        stop2:
        cout << "Would you like to enter new car to database(y or n): ";
fflush(stdin);
        cin >> choice;
        if(choice != 'y' && choice != 'n')
        {
            cout << "\t Invalid input. Try again" << endl;
            goto stop2;
        }

        }while(choice != 'n');
    }
    catch(...)
    {
        cout << "Some technical error occurred, contact developer ASAP."
<< endl;
    }
}

```

```

    system("PAUSE");
}
void adminLogin()
{
    system("CLS");
    char pasd[6] = {'a', 'b', 'c', '1', '2', '3'};
    char inpas[6];
    cout << "\n\n\t\t\t\t\tWELCOME\n\n\n" << endl;
    cout << "\t\t\t\t\t ----- " << endl;
    cout << "\t\t\t\t\t|      ADMIN LOGIN      |" << endl;
    cout << "\t\t\t\t\t ----- " << endl;
    cout << "\t\t\t\t\t \t Password : ";
    for(int i = 0; i < 6; i++)
    {
        inpas[i] = getch(); // a getch() function in a C/ C++ program to hold
        the output screen for some time until the user passes a key from the
        keyboard to exit the console screen.
        cout << "*";
    }
    cout << endl;

    if(pasd[0] == inpas[0] && pasd[1] == inpas[1] && pasd[2] ==
inpas[2] && inpas[3] == pasd[3] && inpas[4] == pasd[4] && inpas[5]
== pasd[5])
    {
        cout << endl << "\t\t\t\t\t Log In successful." << endl;
    }
    else{
        cout << "\t\t\t\t\t Try Again" << endl;
        system("PAUSE");
        // pause() is a C++ method used to pause a program during
        execution. It allows the user to input or read data.
        // The pause approach makes the system more readable and user-
        friendly by allowing the user to read the instructions before performing
        any task.
        adminLogin();
    }
}
void adminPage()
{
    char choice;
    mainPage:
    system("CLS");

```

```

cout << "\t\t CAR RENTAL MANAGEMENT SYSTEM." << endl;
cout << "\t\t -----" << endl;
cout << "\t What would you like to do Sir?" << endl;
cout << "\t 1. Register new car to the Database." << endl;
cout << "\t 2. See all registered cars." << endl;
cout << "\t 3. Customer request for a car." << endl;
cout << "\t 4. Customer wants to return a car." << endl;
cout << "\t 5. See all ongoing trips." << endl;
cout << "\t 6. Delete a car from database." << endl;
cout << "\t 7. LogOut." << endl;
cout << endl << endl;
cout << "\t Enter your choice :"; fflush(stdin); cin >> choice;
switch(choice)
{
    case '1': addCar(); break;
    case '2': displaydata('*'); break;
    case '3': requestcar(); break;
    case '4': returncar(); break;
    case '5': showtrips(); break;
    case '6': deletecar(); break;
    case '7': adminLogin(); break;
    default: cout << "Invalid input" << endl;cout << "Try again." <<
endl;system("PAUSE");
}

    goto mainPage;

}

int main()
{
    adminLogin();
    adminPage();
    return 0;
}

```

CONCLUSION

A car rental management system is a software application designed to streamline the management of a car rental business. The system allows the rental company to manage customer reservations, vehicle inventory, rental contracts, and in a centralized location. This type of system typically includes features such as vehicle tracking, customer management, booking management .

The car rental management system project will involve the development of a software application that incorporates these features. The system should be designed to be user-friendly and efficient, allowing rental company employees to easily access and update customer and vehicle information, manage bookings .

The project will likely involve several key components, including a database to store customer and vehicle information, a user interface to facilitate employee access to the system, and algorithms to manage bookings and rental contracts. The project may also involve the development of reporting tools to help the rental company analyze business performance and identify areas for improvement.

Overall, the goal of the car rental management system project is to create a comprehensive software application that streamlines the rental process and makes it easier for the rental company to manage its operations. The project will likely require collaboration between developers, designers, and business analysts to ensure that the final product meets the needs of the rental company and its customers.