

Computer Science Project

VEHICLE EMISSION RECORDING SYSTEM

By

Abhiram Ravi (10)

Aryan Sureka (18)

INDEX

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Content** | **Page Number** |
| 1 | [Introduction](#_jh46odax4otb) | 2 |
| 2 | [Acknowledgements](#_ozo8b01e3mj3) | 3 |
| 3 | [System Requirement](#_epqqfgjdn6bj) | 4 |
| 4 | [Algorithm](#_anebhswb0tvt) | 5 |
| 5 | [Header Files](#_6igw1vdf75yg) | 6 |
| 6 | [User Defined Functions](#_ec8ectygxrn2) | 7 |
| 7 | [Data Members/Variables](#_u7id99olfuhi) | 8 |
| 8 | [Source Code](#_gf1v0gwv6h53) | 9 |
| 9 | [Screenshots](#_r81keynvqsf9) | 19 |
| 10 | [Bibliography](#_bqdnqcvgkbtm) | 29 |

Introduction

Our project is a vehicle emission recording system which can be used in garages and factories where emission records of the vehicles are to be kept.

A user can log in either as an admin or a guest user. The admin can create, add, modify, delete and view the records and needs to enter a password to do so. The year and the number of vehicles of each type are entered by the user and recorded by the system. The guest user can only view the existing records. The data is saved so it can be retrieved even if the system is turned off. This project can be used to record the emissions for government reports and for research purposes.

Acknowledgements

I would like to express my special thanks of gratitude to my teachers (Mrs Suguna, Mrs Manjula and Mrs Nupur) as well as our principal who gave me this golden opportunity to do this wonderful project on the topic (Vehicle Emission Recording System), which also helped me in doing a lot of research and learn a lot of new things. I am really thankful to them.

Secondly, I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

System Requirement

1 GB RAM (2 GB+ recommended)

9-58 GB free hard disk space depending on edition and configuration, including space required for temporary files

DVD-ROM drive (if installing from a Media Kit DVD)

Basic GPU – Any vendor DirectX 9.0 class or better (Pixel Shader Level 2)

Intel® Pentium® or compatible, 1.6 GHz minimum (2GHz+ recommended)

1024x768 or higher-resolution monitor

Mouse or other pointing device

Microsoft® Windows 8 or 8.1 (32-bit and 64-bit)

Microsoft® Windows 7 SP1 (32-bit and 64-bit)

Microsoft® Windows Server® 2008 and 2012 (32-bit and 64-bit)

C++Builder can also be run on OS X by using a virtual machine (VM) VMware Fusion or Parallels hosting Windows Vista, 7 or 8

Algorithm

The program starts by giving instructions to the user on how to navigate and run the program and displays a menu.

The user can log in as an admin or a guest user. The admin has to enter the password: “password” to enter.

The admin gets a menu of possible operations such as Creating new records, adding new data to existing records, modifying the data in the current records and deleting records.

The guest user can only view the existing records and cannot add new records or modify the existing records.

Header Files

#include<iostream.h>

#include <iomanip.h>

#include<conio.h>

#include<fstream.h>

#include<dos.h>

#include<graphics.h>

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

#include<string.h>

#include <process.h>

User Defined Functions

|  |  |
| --- | --- |
| FUNCTION NAME | PURPOSE |
| void calc() | Calculate total emission |
| getTotem() | Returns total emission |
| getYear() | Returns year |
| getdata() | Input data from user |
| display() | Displays the records |
| TableHeader() | Formatting the menu |
| createRecords() | Writing data to files |
| displayRecords() | Retrieve data from files |
| appendRecords() | Add data to files |
| modifyRecords() | Modify data in the files |
| deleteRecords() | Delete records in the files |
| usermenu() | Displays the usermenu |
| adminmenu() | Displays the menu for the admin |
| guestmenu() | Displays the menu for the user |

Data Members/Variables

|  |  |
| --- | --- |
| DATATYPE | VARIABLE NAME |
| char | Year, choice,  y, ch, pass, adminOrGuest, innerchoice |
| int | Count,  NoOfRecords, flag, gmode,j |
| class | YearlyRecord |
| Fstream objects | fout, fin |
| double | Emmrates, totem |
| unsigned long | totvehicles,em2,em3,em4,emh |
| long | n2,n3,n4,nh |

Source Code

#include<iostream.h>

#include <iomanip.h>

#include<conio.h>

#include<fstream.h>

#include<dos.h>

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

#include<string.h>

#include <process.h>

int NoOfRecords;

double emmrates[4]={0.02,0.03,0.04,0.05};

class YearlyRecord

{

unsigned long totvehicles,em2,em3,em4,emh;

double totem;

char year[5];

long n2,n3,n4,nh;

void calc()

{

totvehicles = n2+n3+n4+nh;

em2 = n2;

em3 = n3;

em4 = n4;

emh = nh;

totem= em2\*emmrates[0] + em3\*emmrates[1] + em4\*emmrates[2] + emh\*emmrates[3];

}

public:

double getTotem() {

return totem;

}

char\* getYear() {

return year;

}

void getdata()

{

cout << endl;

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout << " | Year : : |\b\b\b\b\b\b\b\b";

cin >> year;

cout << " | Number of 2 wheelers : |\b\b\b\b\b\b\b\b";

cin >> n2;

cout << " | Number of 3 wheelers : |\b\b\b\b\b\b\b\b";

cin >> n3;

cout << " | Number of 4 wheelers : |\b\b\b\b\b\b\b\b";

cin >> n4;

cout << " | Number of heavy vehicles : |\b\b\b\b\b\b\b\b";

cin >> nh;

cout << " |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n";

if(year[0]== '-' || strlen(year) > 4 || (n2 < 0) ||( n3 < 0) || (n4 < 0) ||( nh < 0) )

{

if(year[0]== '-' || strlen(year) > 4) {

cout << "\n\n Invalid Data format in record.\n Please enter YEAR only as a 4 digit and positive number.";

}

if((n2 < 0) ||( n3 < 0) || (n4 < 0) ||( nh < 0)) {

cout << "\n\n Invalid Data format in record.\n Please enter NUMBER OF VEHICLES as positive values.";

}

getch();

clrscr();

getdata();

}

calc();

}

void display()

{

cout << " | " << setw(4) << year << " | "<< setw(5) << n2 << " | "<< setw(5) << n3 << " | " << setw(5) << n4 << " | " << setw(5) << nh << " | " << setw(10) << totvehicles << " | " << setw(7) << totem << " |\n";

}

}rec;

void TableHeader() {

clrscr();

cout << " Table of Records - Vehicles and Emmisions\n";

cout << " Emission Rates :\n2 Wheelers - "<<emmrates[0]<<"\n3 Wheelers - "<<emmrates[1]<<"\n4 Wheelers - "<<emmrates[2]<<"\nHeavy Vehicles - "<<emmrates[3]<<"\n";

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

cout << "\n | Year | # of vehicles | total # of vehicles | Emm |\n";

cout << " |-------------------------------------------------------------------------|\n";

cout << " | | 2 wh | 3 wh | 4 wh | heavy | | |\n";

cout << " |-------------------------------------------------------------------------|\n";

}

void createRecords() {

char choice='y';

int count = 1;

ofstream fout;

ifstream fin;

NoOfRecords = 0;

fout.open("Records.dat",ios::out|ios::binary);

while(choice=='y')

{

clrscr();

cout<<"\n Record : " << count << endl;

rec.getdata();

count++;

NoOfRecords++;

fout.write((char\*)&rec,sizeof(rec));

cout<<"\n\n Enter more records ? [y = yes] [n = no] : ";

cin>>choice;

}

ofstream o("RECNT.txt", ios::out);

o << NoOfRecords;

o.close();

getch();

}

void displayRecords() {

ifstream fin;

fin.open("Records.dat", ios::in | ios::binary );

TableHeader();

while(!fin.eof()) {

fin.read((char \*)&rec, sizeof(rec));

if(fin.eof()) break;

rec.display();

}

fin.close();

getch();

}

void appendRecords() {

clrscr();

ifstream i("RECNT.txt", ios::in);

i >> NoOfRecords;

i.close();

ofstream fout;

fout.open("Records.dat", ios :: app | ios :: binary);

YearlyRecord rec;

cout<<"\n Record : " << NoOfRecords + 1 << endl;

rec.getdata();

fout.write((char \*)&rec, sizeof(rec));

NoOfRecords++;

ofstream o("RECNT.txt", ios::out);

o << NoOfRecords;

o.close();

getch();

}

void modifyRecords() {

clrscr();

cout << "\n DISPLAYING ALL EXISTING RECORDS.\n";

displayRecords();

cout << "\n\n ENTER YEAR WHOSE RECORD IS TO BE MODIFIED.\n\n";

cout << " YEAR : ";

char y[4];

gets(y);

YearlyRecord r;

ofstream fout;

fout.open("tmpdata.dat", ios::out | ios::binary );

ifstream fin;

fin.open("Records.dat", ios::in | ios::binary );

while(!fin.eof()) {

fin.read((char \*)&rec, sizeof(rec));

if(fin.eof()) break;

if(strcmp(rec.getYear(), y) == 0) {

clrscr();

cout << " NEW RECORD :\n\n";

r.getdata();

fout.write((char \*)&r, sizeof(r));

}

else {

fout.write((char \*)&rec, sizeof(rec));

}

}

fin.close();

fout.close();

remove("Records.dat");

rename("tmpdata.dat","Records.dat");

getch();

}

void deleteRecords() {

int flag=1;

ifstream i("RECNT.txt", ios::in);

i >> NoOfRecords;

i.close();

clrscr();

cout << "\n DISPLAYING ALL EXISTING RECORDS.\n";

displayRecords();

cout << "\n\n ENTER YEAR WHOSE RECORD IS TO BE DELETED.\n\n";

cout << " YEAR : ";

char y[4];

gets(y);

YearlyRecord r;

ofstream fout;

fout.open("tmpdata.dat", ios::out | ios::binary );

ifstream fin;

fin.open("Records.dat", ios::in | ios::binary );

while(!fin.eof()) {

fin.read((char \*)&rec, sizeof(rec));

if(fin.eof()) break;

if(strcmp(rec.getYear(), y) == 0) {

clrscr();

cout << "SELECTED RECORD:\n\n";

rec.display();

flag=0;

char ch;

cout << "ARE YOU SURE YOU WANT TO DELETE THE SELECTED RECORD ? (Y / N): ";

cin >> ch;

if(!(ch == 'y' || ch == 'Y')) {

fout.write((char \*)&rec, sizeof(rec));

}

else {

NoOfRecords--;

}

}

else {

fout.write((char \*)&rec, sizeof(rec));

}

}

clrscr();

cout<<"\n\n\n\n\n\n\n\n\n";

if(flag==1)

{

cout<<" RECORD NOT FOUND";

}

fin.close();

fout.close();

remove("Records.dat");

rename("tmpdata.dat","Records.dat");

ofstream o("RECNT.txt", ios::out);

o << NoOfRecords;

o.close();

getch();

}

int adminmenu() {

clrscr();

cout << "\n\n\n\n";

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout << " | ADMIN - MENU |\n";

cout << " | |\n";

cout << " | [1] CREATE RECORDS |\n";

cout << " | [2] APPEND RECORDS |\n";

cout << " | [3] MODIFY RECORDS |\n";

cout << " | [4] DELETE RECORDS |\n";

cout << " | [5] DISPLAY RECORDS |\n";

cout << " | [6] EXIT |\n";

cout << " |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n\n";

cout << " ENTER CHOICE < 1 - 6 > : ";

int choice;

cin >> choice;

if(choice > 6 || choice < 1) {

clrscr();

cout << "\n\n\n\n\n\n\n\n\n\n";

cout << " INVALID CHOICE.\n";

cout << " PLEASE RE-ENTER VALID CHOICE\n";

getch();

adminmenu();

}

return choice;

}

int guestmenu() {

clrscr();

cout << "\n\n\n\n\n\n";

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout << " | GUEST - MENU |\n";

cout << " | |\n";

cout << " | [1] DISPLAY RECORDS |\n";

cout << " | [2] EXIT |\n";

cout << " |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n";

cout << endl;

cout << " ENTER CHOICE < 1 - 2 > : ";

int choice;

cin >> choice;

if(choice > 2 || choice < 1) {

clrscr();

cout << "\n\n\n\n\n\n\n\n\n\n";

cout << " INVALID CHOICE.\n";

cout << " PLEASE RE-ENTER VALID CHOICE\n";

getch();

guestmenu();

}

return choice;

}

int usermenu() {

clrscr();

cout << "\n\n\n\n\n\n";

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout << " | USER - SELECTION |\n";

cout << " | |\n";

cout << " | [1] ADMIN |\n";

cout << " | [2] GUEST |\n";

cout << " | [3] EXIT |\n";

cout << " |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n";

cout << endl;

cout << " ENTER USER < 1 - 3 > : ";

int choice;

cin >> choice;

if(choice > 3 || choice < 1) {

clrscr();

cout << "\n\n\n\n\n\n\n\n\n\n";

cout << " INVALID CHOICE.\n";

cout << " PLEASE RE-ENTER VALID CHOICE\n";

getch();

usermenu();

}

return choice;

}

void main()

{

clrscr();

int gmode,j=0;

cout<<"VERS\n";

cout<<"Vehicle Emission Recording System";

cout<<"INSTRUCTIONS\n";

cout<<"Enter only years having 4 numbers.\n";

cout<<"Number of vehicles of any type not to exceed 6 digits.\n";

cout<<"Do not have more than 6 records at any time.\n";

cout<<"press any key to continue.";

char ch;

cin>>ch;

int adminOrGuest, innerchoice;

char pass[10];

do {

adminOrGuest = usermenu();

switch(adminOrGuest)

{

case 1: cout<<"\n ENTER PASSWORD: \n";

cout<<" ";

cin>>pass;

if(strcmp(pass,"password")==0)

{

innerchoice = adminmenu();

switch(innerchoice)

{

case 1: createRecords(); break;

case 2: appendRecords(); break;

case 3: modifyRecords(); break;

case 4: deleteRecords(); break;

case 5: displayRecords(); break;

case 6: exit(0);

}

}

else{

clrscr();

cout << "\n\n\n\n\n";

cout<<" UNAUTHORIZED ACCESS. ACCESS DENIED.\n";

cout<<" ENTER THE RIGHT password TO CONTINUE\n";

getch();

}

break;

case 2: innerchoice = guestmenu();

switch(innerchoice)

{

case 1: displayRecords(); break;

case 2: exit(0);

}

break;

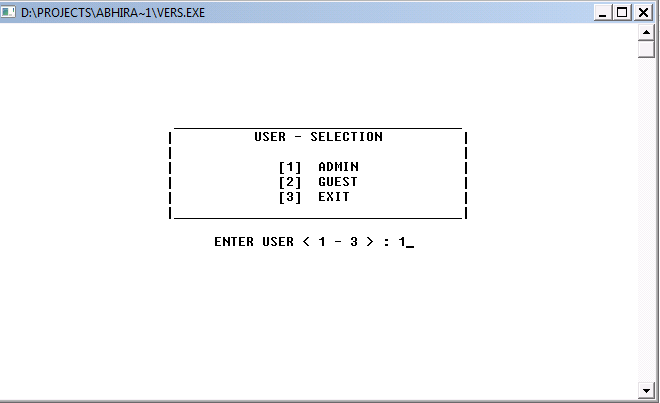
case 3: exit(0);

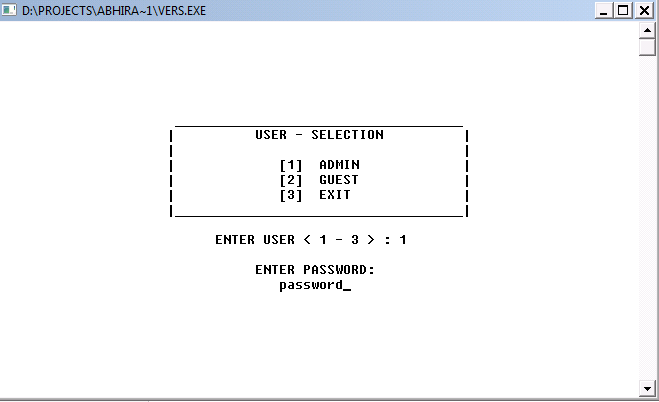
}

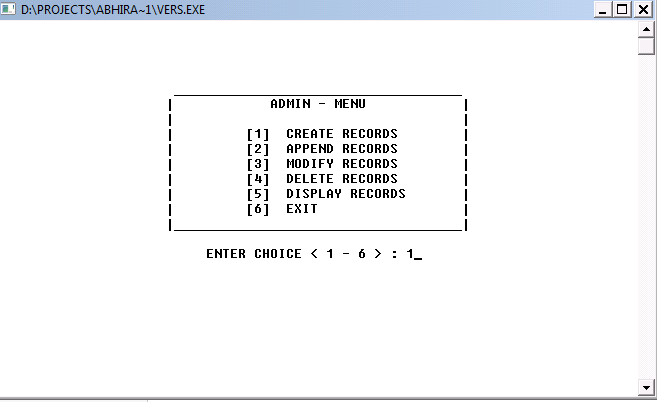
} while(1);

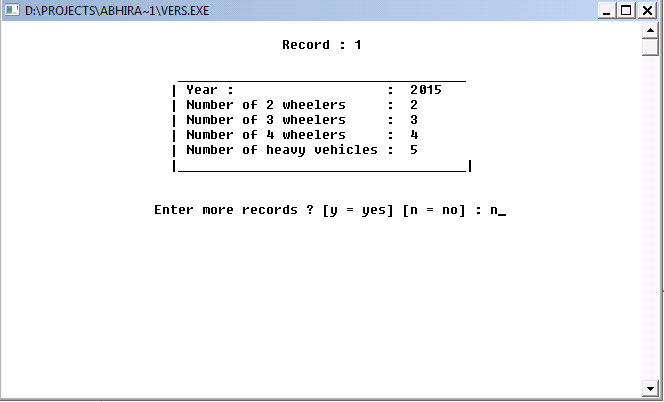
} }

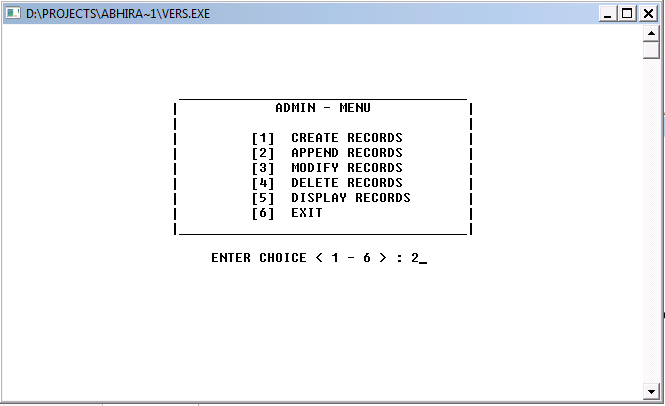
Screenshots

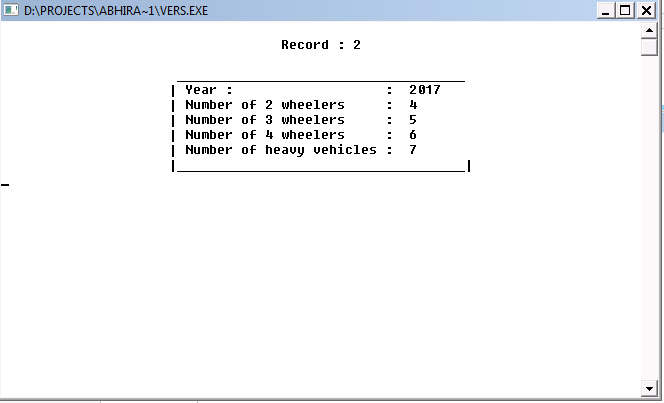


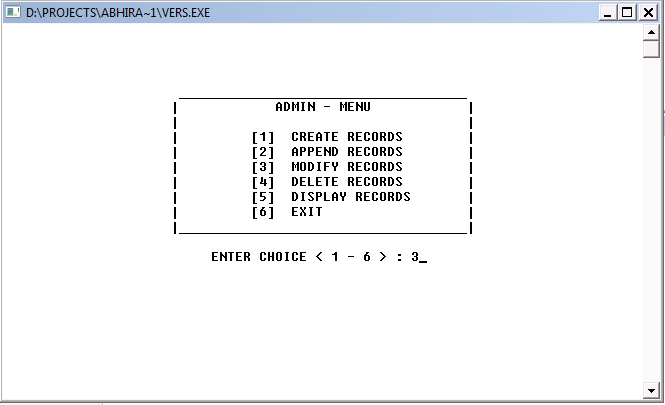


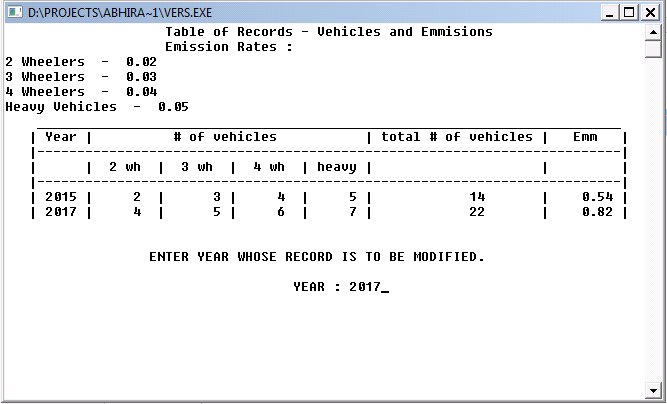


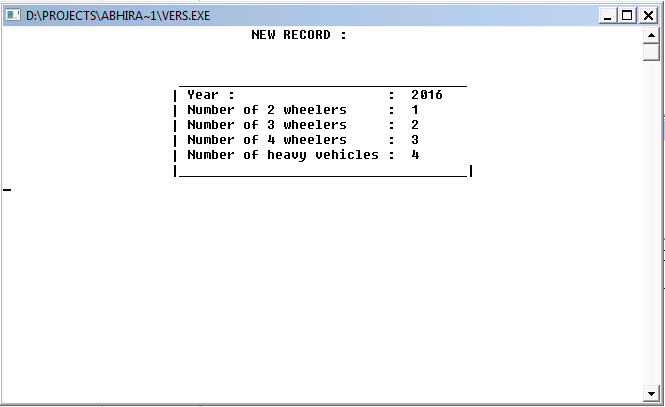


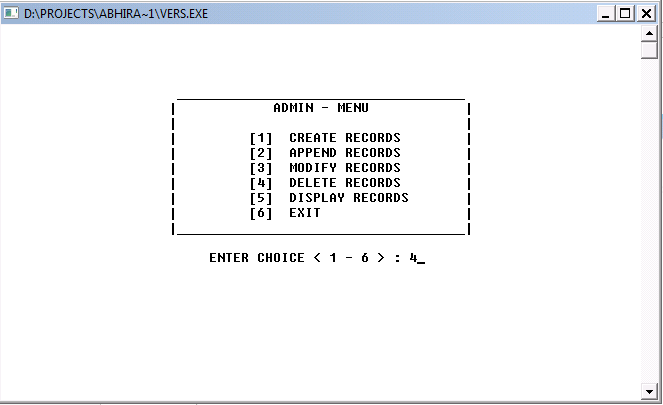


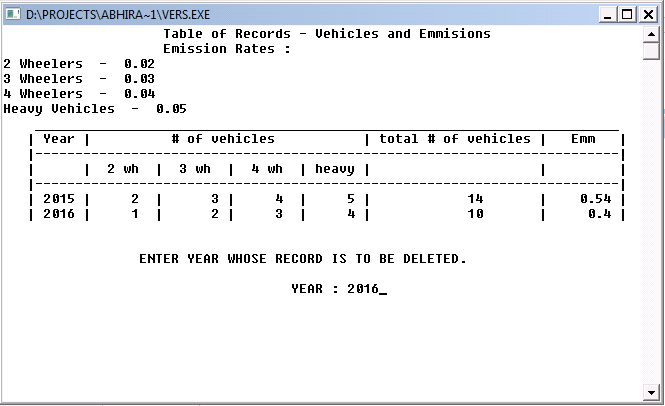


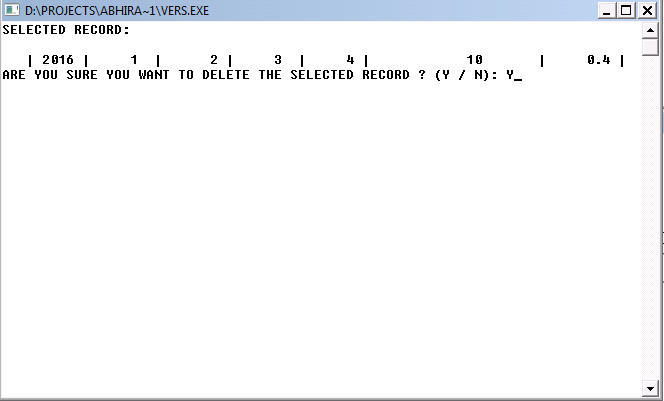


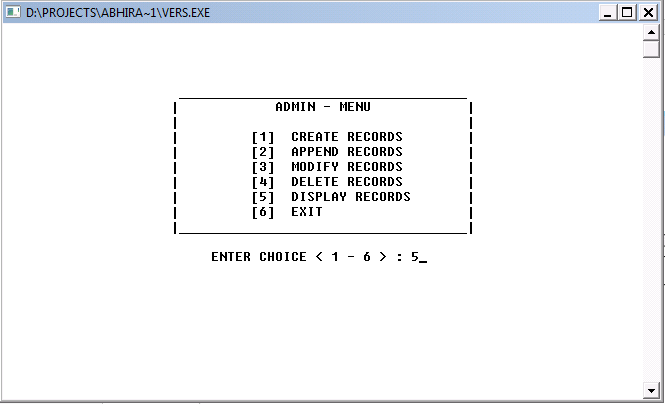


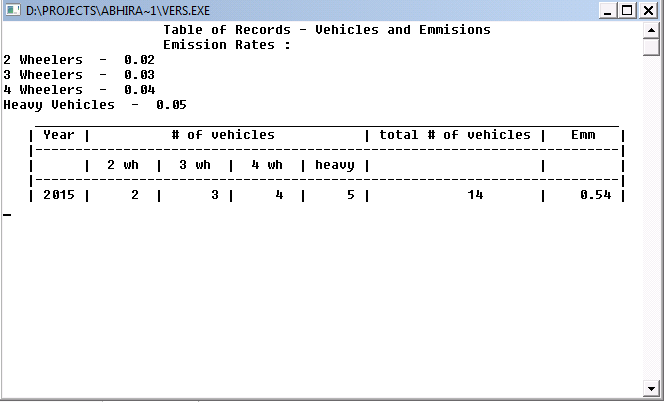


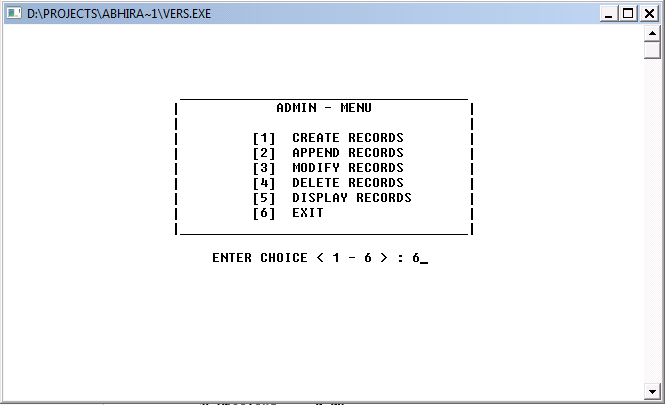


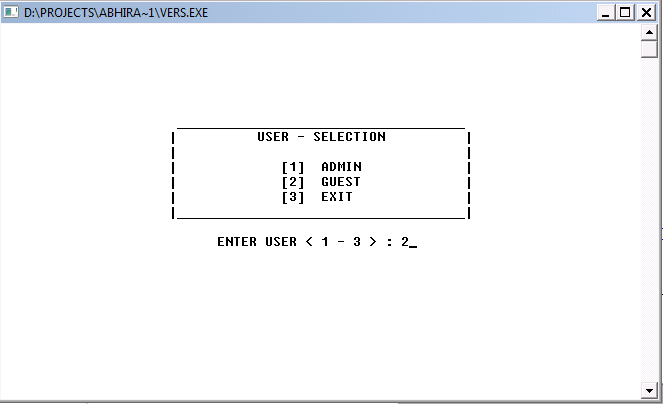


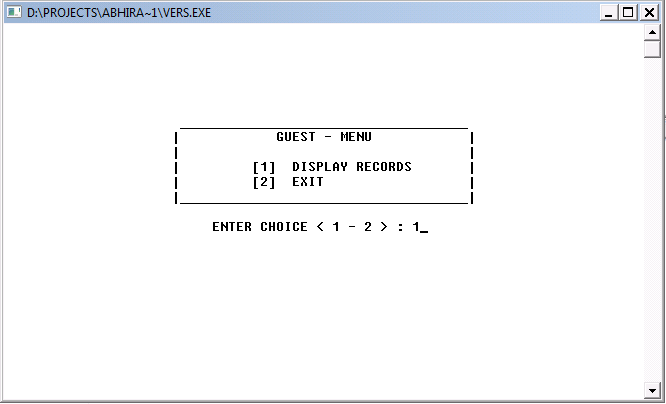


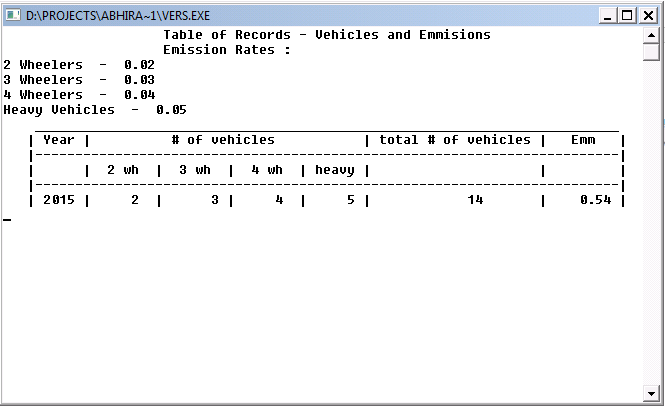


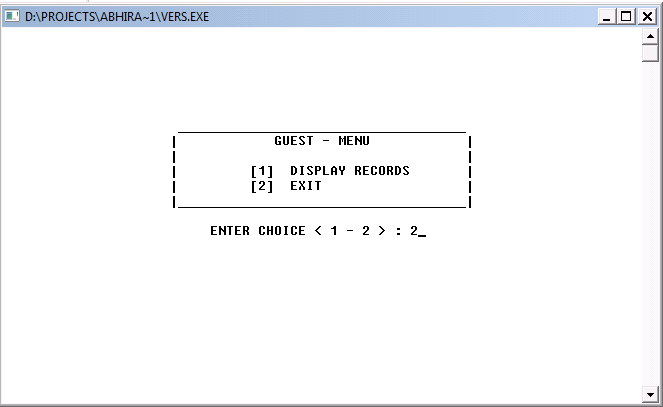












Bibliography

Computer science with C++ for class 11

Computer science with C++ for class 12