NAME=RINKU YADAV

CODE OF =CAR.C

// Include Header Files

#include "car.h"

#include <stdio.h>

#include <string.h>

#include "conio2.h"

// Defination of addAdmin() Function

void addAdmin(){

// Open a admin.bin file in rb mode

FILE \*fp = fopen("admin.bin","rb");

// Check Weather file is open or not

if(fp == NULL){

// Now Open file admin.bin in wb mode

fp = fopen("admin.bin","wb");

// Making array of Stucture of User and it's size is 2

User u[2] = {{"admin","test","Ravi"},{"super","demo","Anil"}};

// Now Write User array content in emp.bin file

fwrite(u,sizeof(u),1,fp);

}

// finally close emp file for save data in this file

fclose(fp);

}

// Defination of getInput() function which have "return type" is "User Pointer"

User \* getInput(){

// Declaring Variable

int i;

// Create Static Variable of User Structure Type , that's why we can return it

// \*\*\*\*\*\* Remember When we want to return address of variable or structure

// then we use STATIC in front of that variable because

// when function is end than , Whole variable of this function is destroyed by compiler

// There is concept of dangling pointer

// But When we use static the variable is live till the end of program

static User u;

// clear screen

clrscr();

// Now we shift our cursor to 32 column and 1 row

gotoxy(32,1);

// We are Changing our text color in output screen

textcolor(YELLOW);

printf("CAR RENTAL SYSTEM\n");

// Run for loop 80 time to create a line in output screen

for(i=1;i<=80;i++){

printf("%c",247);

}

// Again shift our cursor at 32 col and 4 row

gotoxy(32,4);

printf("\* LOGIN PANEL \*");

gotoxy(1,7);

textcolor(LIGHTCYAN);

for(i=1;i<=80;i++){

printf("%c",247);

}

gotoxy(1,15);

for(i=1;i<=80;i++){

printf("%c",247);

}

gotoxy(60,8);

textcolor(WHITE);

printf("Press 0 to exit");

gotoxy(25,10);

textcolor(LIGHTCYAN);

printf("Enter user id :");

// we are cleaning the buffer area

fflush(stdin);

textcolor(WHITE);

// Remember --> fgets is used to take multiple words with control input

fgets(u.userid,20,stdin);

// pos is pointer of char type

char \*pos;

// Remember --> strchr is use to search a character in given string

// it takes two input 1. charater\_array , 2. The word which we want to search in string

// it return --> either address of character or NULL

// if it find character in given string then it return character's address

// if it didn't found character then it return Null

pos = strchr(u.userid,'\n');

// check what kind of value carry by pos after strchr function

if(pos != NULL){

// We are replaceing '\n' by '\0' in given string in strchr() function

\*pos = '\0';

}

// if user entered 0 that's means he/she want to cancel the login

if((strcmp(u.userid,"0"))==0){

textcolor(LIGHTRED);

gotoxy(30,17);

printf("Login Cancelled !");

fflush(stdin);

getch();

return NULL;

}

// we are take password form user in form HIDDEN character

gotoxy(25,11);

textcolor(LIGHTCYAN);

printf("Enter Password :");

fflush(stdin);

i = 0;

for(;;){

u.pwd[i] = getch();

// if check Weather user enter "ENTER" or NOT

if(u.pwd[i] == 13){

u.pwd[i] = '\0';

break;

}

// if check Weather user enter "BACKSPACE" or NOT

if(u.pwd[i]==8){

i-=1;

printf("\b \b");

continue;

}

else{

printf("\*");

i++;

}

}

// TESTING --> printf("%s",u.pwd);

// It check , Is user want to cancel Login

if((strcmp(u.pwd,"0"))==0){

textcolor(LIGHTRED);

gotoxy(30,17);

printf("Login Cancelled !");

fflush(stdin);

getch();

textcolor(YELLOW);

return NULL;

}

//getch();

return &u;

}

// This check Weather user is exist or not which is given as input

// Remember --> with 2nd argument we can decide wheather we have to check admin detail or emp deatil

int checkUserExist(User u,char \* usertype){

// Here we handle if user enter only empty string

if(strlen(u.userid)==0 || strlen(u.pwd)==0){

textcolor(LIGHTRED);

gotoxy(28,20);

printf("Both fields are Mandatory. Try Again !");

getch();

gotoxy(28,20);

// This is technique by which we can erase certain part of output screen

printf("\t\t\t\t\t\t");

return 0;

}

// if string is not empty then user input something , Now we deal with it

FILE \*fp;

// This if decides that whose detail we have to check (admin or emp)

if( (strcmp(usertype,"admin"))==0 ){

fp=fopen("admin.bin","rb");

}else{

fp=fopen("emp.bin","rb");

}

// Handle If file is not open

if(fp == NULL){

gotoxy(28,20);

textcolor(LIGHTRED);

printf("Sorry ! cannot open the file");

getch();

gotoxy(28,20);

printf("\t\t\t\t\t");

return -1;

}

// with the help of found variable we decide recode is match or not

int found = 0;

User user;

// this loop read recodes one by one and check userid with every recode

while(fread(&user,sizeof(user),1,fp) == 1){

// printf("Id -> '%s' Pwd -> '%s'\n",user.userid,user.pwd);

// printf("Id -> '%s' Pwd -> '%s'\n",u.userid,u.pwd);

//getch();

if((strcmp(u.userid,user.userid) == 0) && (strcmp(u.pwd,user.pwd) == 0)){

found = 1;

break;

}

}

// means recode not found

if(found == 0){

gotoxy(28,20);

textcolor(LIGHTRED);

printf("Invalid userid/password . Try Again");

getch();

gotoxy(28,20);

printf("\t\t\t\t\t\t");

}else{ // recode is found

gotoxy(28,20);

textcolor(LIGHTGREEN);

printf("Login successful !");

getch();

gotoxy(28,20);

printf("\t\t\t\t\t\t");

}

fclose(fp);

return found;

}

//This function show all functionality of admin

int adminMenu(){

int i,choice;

textcolor(LIGHTRED);

gotoxy(32,2);

printf("CAR RENTAL SYSTEM");

textcolor(LIGHTGREEN);

gotoxy(35,6);

printf("ADMIN MENU\n");

for(i = 1; i<= 80 ; i++){

printf("\*");

}

textcolor(YELLOW);

gotoxy(32,8);

printf("1. Add Employee");

gotoxy(32,9);

printf("2. Add Car Details");

gotoxy(32,10);

printf("3. Show Employee");

gotoxy(32,11);

printf("4. Show Car Details");

gotoxy(32,12);

printf("5. Delete Employee");

gotoxy(32,13);

printf("6. Delete Car Details");

gotoxy(32,14);

printf("7. Exit");

gotoxy(32,16);

printf("Enter Choice : ");

scanf("%d",&choice);

return choice;

}

// This funtion add Employ which done by Admin

void addEmployee()

{

char empid[10] ={"EMP-"};

char temp[20];

char choice;

char \*pos;

User u;

FILE \*fp = fopen("emp.bin","ab+");

fseek(fp,0,SEEK\_END);

long total\_rec = ftell(fp)/sizeof(User);

if(total\_rec != 0){

fseek(fp,-60,SEEK\_END);

fread(temp,sizeof(temp),1,fp);

pos = strchr(temp,'-');

total\_rec = atoi(pos+1);

}

total\_rec++;

sprintf(temp,"%d",total\_rec);

strcat(empid,temp);

strcpy(u.userid,empid);

fseek(fp,0,SEEK\_END);

do{

clrscr();

int i;

textcolor(LIGHTRED);

gotoxy(32,2);

printf("CAR RENTAL APP");

gotoxy(1,3);

textcolor(LIGHTGREEN);

for(i = 1; i <= 80 ; i++)

printf("~");

gotoxy(25,5);

textcolor(WHITE);

printf("\*\*\*\*\* ADD EMPLOYEE DETAILS \*\*\*\*\*");

gotoxy(1,8);

textcolor(YELLOW);

printf("Enter Employee Name :");

fflush(stdin);

fgets(u.name,20,stdin);

pos =strchr(u.name,'\n');

if(pos != NULL){

\*pos = '\0';

}

textcolor(YELLOW);

printf("Enter Employee Pwd :");

fflush(stdin);

textcolor(WHITE);

fgets(u.pwd,20,stdin);

pos = strchr(u.pwd,'\n');

if(pos != NULL){

\*pos = '\0';

}

fwrite(&u,sizeof(u),1,fp);

gotoxy(30,15);

textcolor(LIGHTGREEN);

printf("EMPLOYEE ADDED SUCCESSFULLY !!!");

printf("\nEMPLOYEE ID is %s",u.userid);

getch();

textcolor(LIGHTRED);

gotoxy(1,20);

printf("Do you want to add more employees (Y/N)?");

textcolor(WHITE);

fflush(stdin);

scanf("%c",&choice);

if(choice == 'N' || choice == 'n')

break;

total\_rec++;

sprintf(u.userid,"EMP-%d",total\_rec);

}while(1);

fclose(fp);

}

// this function simply add a car

void addCarDetails()

{

FILE \* fp = fopen("car.bin","ab+");

Car c;

long total\_car;

if(fp == NULL){

//Messege of not open this file

}

/\*while(fread(&c,sizeof(c),1,fp) == 1){

printf("\n%d\n",ftell(fp));

getch();

}\*/

fseek(fp,0,SEEK\_END);

total\_car = ftell(fp)/sizeof(c);

int last\_car\_id;

if(total\_car != 0){

fseek(fp,-68,SEEK\_END);

fread(&c,sizeof(c),1,fp);

last\_car\_id = c.car\_id;

total\_car = last\_car\_id;

}

total\_car++;

do{

int i; // for loop --> display ~~~~~

char ch;

clrscr();

c.car\_id = total\_car;

gotoxy(32,2);

textcolor(LIGHTRED);

printf("CAR RENTAL APP\n");

textcolor(LIGHTGREEN);

for( i = 1 ; i<= 80 ; i++)

printf("~");

gotoxy(25,5);

textcolor(WHITE);

printf("\*\*\*\*\* ADD CAR DETAILS \*\*\*\*\*");

gotoxy(1,8);

textcolor(YELLOW);

printf("Enter Car Model : ");

fflush(stdin);

textcolor(WHITE);

// Take input from admin/emp

fgets(&c.car\_name,50,stdin);

char \*pos;

pos = strchr(c.car\_name,'\n');

if(pos != NULL){

pos == '\0';

}

// <--- car capicity --->

textcolor(YELLOW);

printf("Enter Car Capicity : ");

textcolor(WHITE);

scanf("%d",&c.capacity);

// <--- car count --->

textcolor(YELLOW);

printf("Enter Car Count : ");

textcolor(WHITE);

scanf("%d",&c.car\_count);

// <--- price --->

textcolor(YELLOW);

printf("Enter Car Price for per Day : ");

textcolor(WHITE);

scanf("%d",&c.price);

// <--- Messege of Successfully add car --->

textcolor(LIGHTGREEN);

gotoxy(30,15);

printf("CAR ADD SUCCESSFULLY");

printf("\nCAR ID IS : %d",c.car\_id);

gotoxy(1,20);

textcolor(LIGHTRED);

// if you remove record in write in file

fseek(fp,0,SEEK\_END);

fwrite(&c,sizeof(c),1,fp);

printf("DO YOU WANT TO ADD MORE CAR (Y/N) : ");

fflush(stdin);

scanf("%c",&ch);

if(ch == 'N' || ch == 'n')

break;

total\_car++;

}while(1);

fclose(fp);

}

void viewEmployee(){

clrscr();

int i,row;

textcolor(YELLOW);

gotoxy(32,1);

printf("CAR RENTAL SYSTEM");

gotoxy(1,2);

for(i=1;i<=80;i++)

printf("%c",247);

gotoxy(31,5);

printf("\* EMPLOYEE DETAILS \*");

textcolor(LIGHTGREEN);

gotoxy(1,7);

for(i=1;i<=80;i++)

printf("%c",247);

FILE \* fp = fopen("emp.bin","rb");

if(fp == NULL){

gotoxy(27,9);

textcolor(LIGHTRED);

printf("Sorry ! No Employee added yet.");

getch();

return;

}

printf("Employee ID\t\t\tName\t\t\tPassword");

gotoxy(1,9);

for(i=1;i<=80;i++)

printf("%c",247);

User u;

row = 10;

fseek(fp,0,SEEK\_SET);

while(fread(&u,sizeof(u),1,fp)==1){

gotoxy(1,row);

printf("%s",u.userid);

gotoxy(33,row);

printf("%s",u.name);

gotoxy(57,row);

printf("%s",u.pwd);

row++;

}

fclose(fp);

getch();

}

void showCarDetails(){

clrscr();

int i,row;

textcolor(YELLOW);

gotoxy(32,1);

printf("CAR RENTAL SYSTEM");

gotoxy(1,2);

for(i=1;i<=80;i++)

printf("%c",247);

gotoxy(31,5);

printf("\* ALL CAR DETAILS \*");

textcolor(LIGHTGREEN);

gotoxy(1,7);

for(i=1;i<=80;i++)

printf("%c",247);

FILE \* fp = fopen("car.bin","rb");

if(fp == NULL){

gotoxy(27,11);

textcolor(LIGHTRED);

printf("Sorry ! No Car added yet.");

getch();

return;

}

printf("CAR ID\t\tModel\t\tCapicity\tAvailable\tPrice/Day");

gotoxy(1,9);

for(i=1;i<=80;i++)

printf("%c",247);

Car c;

row = 10;

fseek(fp,0,SEEK\_SET);

while(fread(&c,sizeof(c),1,fp)==1){

gotoxy(1,row);

printf("%d",c.car\_id);

gotoxy(17,row);

printf("%s",c.car\_name);

gotoxy(33,row);

printf("%d",c.capacity);

gotoxy(49,row);

printf("%d",c.car\_count);

gotoxy(65,row);

printf("%d",c.price);

row++;

}

fclose(fp);

getch();

}

int deleteEmp()

{

FILE \*fp1 = fopen("emp.bin","rb");

int i,result;

char empid[20];

textcolor(YELLOW);

gotoxy(32,1);

printf("CAR RENTAL SYSTEM");

gotoxy(1,2);

for(i = 1 ; i <= 80 ; i++)

printf("%c",247);

gotoxy(27,5);

printf("\* DELETE EMPLOYEE RECORD \*");

textcolor(LIGHTGREEN);

gotoxy(1,7);

for(i = 1; i<= 80 ; i++)

printf("%c",247);

gotoxy(1,12);

for( i = 1 ; i <= 80 ; i++)

printf("%c",247);

if(fp1 == NULL){

textcolor(LIGHTRED);

gotoxy(30,9);

printf("No Employee Added Yet!");

return -1;

}

FILE \*fp2 = fopen("temp.bin","wb+");

gotoxy(10,9);

textcolor(YELLOW);

printf("Enter Employee Id to delete the record :");

textcolor(WHITE);

fflush(stdin);

scanf("%s",empid);

User u;

int found = 0;

while(fread(&u,sizeof(u),1,fp1) == 1){

if(strcmp(u.userid,empid) != 0)

fwrite(&u,sizeof(u),1,fp2);

else

found = 1;

}

fclose(fp1);

fclose(fp2);

if(found == 0)

remove("temp.bin");

else{

result = remove("emp.bin");

if(result != 0 )

return 2;

result = rename("temp.bin","emp.bin");

if(result != 0)

return 2;

}

return found;

}

int deleteCarModel()

{

FILE \*fp1 = fopen("car.bin","rb");

int i,result;

int carid;

textcolor(YELLOW);

gotoxy(32,1);

printf("CAR RENTAL SYSTEM");

// For Designing

gotoxy(1,2);

for(i = 1 ; i <= 80 ; i++)

printf("%c",247);

// For Desingning

gotoxy(29,5);

printf("\* DELETE CAR MODEL \*");

// For Desingning

textcolor(LIGHTGREEN);

gotoxy(1,7);

for(i = 1; i<= 80 ; i++)

printf("%c",247);

// For Desingning

gotoxy(1,12);

for( i = 1 ; i <= 80 ; i++)

printf("%c",247);

// check condition weather file is open / exist or not

if(fp1 == NULL){

textcolor(LIGHTRED);

gotoxy(32,9);

printf("No Car Added Yet!");

return -1;

}

// if file fp1 is successfully open then create a temp file

FILE \*fp2 = fopen("temp.bin","wb+");

// take car id

gotoxy(10,9);

textcolor(YELLOW);

printf("Enter Car Id to delete the record :");

textcolor(WHITE);

scanf("%d",&carid);

Car c;

// if car is found then set found = 1

int found = 0;

// loop for read all record from car.bin file

while(fread(&c,sizeof(c),1,fp1) == 1){

// this is condition to check carid is same or not

if(carid != c.car\_id)

// if carid is not match then simply write car detail in temp

fwrite(&c,sizeof(c),1,fp2);

else

// if carid is match then set found = 1 , to determine car is found or not

found = 1;

}

// finally close both files

fclose(fp1);

fclose(fp2);

// delete temp/car and rename

if(found == 0)

remove("temp.bin");

else{

result = remove("car.bin");

if(result != 0 )

return 2;

result = rename("temp.bin","car.bin");

if(result != 0)

return 2;

}

return found;

}

// Show Employee Menu

int empMenu()

{

int choice,i;

textcolor(LIGHTRED);

gotoxy(32,2);

printf("CAR RENTAL SYSTEM");

textcolor(LIGHTGREEN);

gotoxy(34,6);

printf("EMPLOYEE MENU\n");

for(i = 1 ; i <= 80 ; i++)

printf("\*");

gotoxy(32,8);

printf("1. Rent A CAR");

gotoxy(32,9);

printf("2. Booking Details");

gotoxy(32,10);

printf("3. Available Car Details");

gotoxy(32,11);

printf("4. Show All Car Details");

gotoxy(32,12);

printf("5. Exit");

gotoxy(32,15);

printf("Enter Choice : ");

scanf("%d",&choice);

return choice;

}

int selectCarModel()

{

FILE \*fp = fopen("car.bin","rb");

int flag;

int choice,rowno = 9;

Car C;

gotoxy(34,rowno);

int carcount = 0;

while(fread(&C,sizeof(C),1,fp) == 1){

if(C.car\_count > 0)

{

printf("%d . %s",C.car\_id,C.car\_name);

gotoxy(34,++rowno);

++carcount;

}

}

if(carcount == 0)

return -2;

gotoxy(34,rowno + 2);

printf("Enter Your Choice(0 to quit) : ");

while(1)

{

flag = 0;

scanf("%d",&choice);

if(choice == 0)

{

fclose(fp);

return 0;

}

rewind(fp);

while(fread(&C,sizeof(C),1,fp) == 1)

{

if(C.car\_id == choice && C.car\_count > 0)

{

flag = 1;

break;

}

}

if(flag == 1)

{

fclose(fp);

return flag;

}

gotoxy(37,rowno + 4);

textcolor(LIGHTRED);

printf("Wrong Input");

getch();

gotoxy(37,rowno + 4);

printf("\t\t\t");

gotoxy(53,rowno + 2);

printf("\t\t\t");

gotoxy(53,rowno + 2);

textcolor(WHITE);

}

}

char \* getCarName(int c\_id)

{

FILE \* fp = fopen("car.bin","rb");

if(fp==NULL)

{

printf("Sorry ! File cannot be opened");

return NULL;

}

static Car C;

while(fread(&C,sizeof(C),1,fp) == 1)

{

if(C.car\_id == c\_id)

{

break;

}

fclose(fp);

}

return C.car\_name;

}

int isValidDate(struct tm dt)

{

if(dt.tm\_year >= 2021 && dt.tm\_year <= 2022)

{

if(dt.tm\_mon >= 1 && dt.tm\_mon <=12)

{

if((dt.tm\_mday >= 1 && dt.tm\_mday <= 31) && (dt.tm\_mon == 1 || dt.tm\_mon == 3 || dt.tm\_mon == 5 || dt.tm\_mon == 7 || dt.tm\_mon == 8 || dt.tm\_mon == 10 || dt.tm\_mon == 12))

{

return 1;

}

else if((dt.tm\_mday >= 1 && dt.tm\_mday <= 30) && (dt.tm\_mon == 4 || dt.tm\_mon == 6 || dt.tm\_mon == 9 || dt.tm\_mon == 11))

{

return 1;

}

else if((dt.tm\_mday >= 1 && dt.tm\_mday <= 28) && (dt.tm\_mon == 2 ))

{

return 1;

}

else if((dt.tm\_mday >= 1 && dt.tm\_mday <= 29) && (dt.tm\_mon == 2) && (dt.tm\_year % 400 ||(dt.tm\_year % 4==0 && dt.tm\_year % 100 != 0)))

{

return 1;

}

}

}

return 0;

}

int rentCar()

{

Customer\_Car\_Details CC;

int c,i;

return\_car();

return\_car\_delete();

textcolor(LIGHTRED);

gotoxy(32,2);

printf("CAR RENTAL SYSTEM\n");

textcolor(LIGHTGREEN);

gotoxy(35,6);

printf("EMPLOYEE MENU\n");

for(i = 1; i <= 80 ; i++)

printf("\*");

textcolor(YELLOW);

gotoxy(32,8);

c = selectCarModel();

if(c == 0)

return 0;

clrscr();

CC.car\_id = c;

textcolor(LIGHTRED);

gotoxy(32,2);

printf("CAR RENTAL SYSTEM\n");

textcolor(LIGHTGREEN);

gotoxy(35,6);

printf("EMPLOYEE MENU\n");

for(i = 1; i <= 80 ; i++)

printf("\*");

gotoxy(1,17);

textcolor(YELLOW);

gotoxy(27,9);

printf("Enter Customer Name:");

fflush(stdin);

textcolor(WHITE);

fgets(CC.cust\_name,30,stdin);

char \*pos;

pos = strchr(CC.cust\_name,'\n');

if(pos != NULL)

\*pos = '\0';

textcolor(YELLOW);

gotoxy(27,10);

printf("Enter Pickup Point:");

textcolor(WHITE);

fgets(CC.pick,30,stdin);

pos = strchr(CC.pick,'\n');

if(pos != NULL)

\*pos = '\0';

textcolor(YELLOW);

gotoxy(27,11);

printf("Enter drop point:");

textcolor(WHITE);

fgets(CC.drop,30,stdin);

pos = strchr(CC.drop,'\n');

if(pos != NULL)

\*pos = '\0';

gotoxy(27,12);

textcolor(YELLOW);

printf("Enter start date(dd/mm/yyyy):");

textcolor(WHITE);

do

{

scanf("%d %d %d",&CC.sd.tm\_mday,&CC.sd.tm\_mon,&CC.sd.tm\_year);

int datevalid = isValidDate(CC.sd);

if(datevalid == 1)

break;

gotoxy(27,18);

textcolor(LIGHTRED);

printf("Wrong Date!");

getch();

gotoxy(27,18);

printf("\t\t\t");

gotoxy(56,12);

printf("\t\t\t");

gotoxy(56,12);

textcolor(WHITE);

}while(1);

textcolor(YELLOW);

gotoxy(27,13);

printf("Enter end date(dd/mm/yyyy):");

textcolor(WHITE);

do

{

scanf("%d %d %d",&CC.ed.tm\_mday,&CC.ed.tm\_mon,&CC.ed.tm\_year);

int datevalid = isValidDate(CC.sd);

if(datevalid == 1)

break;

gotoxy(27,18);

textcolor(LIGHTRED);

printf("Wrong Date!");

getch();

gotoxy(27,18);

printf("\t\t\t");

gotoxy(54,13);

printf("\t\t\t");

gotoxy(54,13);

textcolor(WHITE);

}while(1);

int valid = checkDate(CC.sd,CC.ed);

if(valid != 1)

{

while(valid != 1)

{

gotoxy(20,22);

textcolor(LIGHTRED);

printf("!!! Booked Date must be Greater Than Current Date !!!");

gotoxy(23,23);

printf("!!! End Date must be Greater Than Start Date !!!");

getch();

gotoxy(20,22);

printf("\t\t\t\t\t\t\t\t\t");

gotoxy(23,23);

printf("\t\t\t\t\t\t\t\t\t");

gotoxy(27,12);

printf("\t\t\t\t\t");

gotoxy(27,13);

printf("\t\t\t\t\t");

gotoxy(27,12);

textcolor(YELLOW);

printf("Enter start date(dd/mm/yyyy):");

textcolor(WHITE);

do

{

scanf("%d %d %d",&CC.sd.tm\_mday,&CC.sd.tm\_mon,&CC.sd.tm\_year);

int datevalid = isValidDate(CC.sd);

if(datevalid == 1)

break;

gotoxy(27,18);

textcolor(LIGHTRED);

printf("Wrong Date!");

getch();

gotoxy(27,18);

printf("\t\t\t");

gotoxy(56,12);

printf("\t\t\t");

gotoxy(56,12);

textcolor(WHITE);

}while(1);

textcolor(YELLOW);

gotoxy(27,13);

printf("Enter end date(dd/mm/yyyy):");

textcolor(WHITE);

do

{

scanf("%d %d %d",&CC.ed.tm\_mday,&CC.ed.tm\_mon,&CC.ed.tm\_year);

int datevalid = isValidDate(CC.sd);

if(datevalid == 1)

break;

gotoxy(27,18);

textcolor(LIGHTRED);

printf("Wrong Date!");

getch();

gotoxy(27,18);

printf("\t\t\t");

gotoxy(54,13);

printf("\t\t\t");

gotoxy(54,13);

textcolor(WHITE);

}while(1);

valid = checkDate(CC.sd,CC.ed);

}

}

FILE \*fp;

fp = fopen("customer.bin","ab");

if(fp == NULL)

{

gotoxy(27,18);

printf("Sorry! File cannot be opened");

return -1;

}

fwrite(&CC,sizeof(Customer\_Car\_Details),1,fp);

gotoxy(27,18);

textcolor(WHITE);

printf("Booking Done!");

printf("\nPress any key to continue");

getch();

updateCarCount(CC.car\_id);

fclose(fp);

return 1;

}

void availCarDetails()

{

return\_car();

return\_car\_delete();

clrscr();

FILE \* fp = fopen("car.bin","rb");

Car C;

int i;

textcolor(YELLOW);

gotoxy(32,1);

printf("CAR RENTAL SYSTEM\n");

textcolor(LIGHTGREEN);

for(i = 1 ; i<= 80; i++)

printf("%c",247);

gotoxy(32,5);

textcolor(YELLOW);

printf("\* AVAILABLE CAR DETAILS \*");

gotoxy(1,7);

textcolor(LIGHTGREEN);

for ( i = 1; i <= 80; i++)

printf("%c",247);

if(fp == NULL)

{

gotoxy(32,8);

textcolor(LIGHTRED);

printf("Sorry! File cannot be opened!");

getch();

return;

}

gotoxy(1,8);

printf("Car ID");

gotoxy(15,8);

printf("Model");

gotoxy(35,8);

printf("Capacity");

gotoxy(50,8);

printf("Availabe");

gotoxy(65,8);

printf("Price/Day");

gotoxy(1,9);

for(i = 1; i<= 80; i++)

printf("%c",247);

int row = 10;

textcolor(YELLOW);

while(fread(&C,sizeof(C),1,fp) == 1)

{

if(C.car\_count > 0)

{

gotoxy(2,row);

printf("%d",C.car\_id);

gotoxy(15,row);

printf("%s",C.car\_name);

gotoxy(35,row);

printf("%d",C.capacity);

gotoxy(50,row);

printf("%d",C.car\_count);

gotoxy(65,row);

printf("%d",C.price);

row++;

}

}

fclose(fp);

getch();

}

void updateCarCount(int c\_id)

{

FILE \* fp = fopen("car.bin","rb+");

if(fp == NULL)

{

textcolor(LIGHTRED);

return ;

}

Car C;

while(fread(&C,sizeof(C),1,fp) == 1)

{

if(C.car\_id == c\_id)

{

int x = C.car\_count;

x--;

fseek(fp,-8,SEEK\_CUR);

fwrite(&x,sizeof(int),1,fp);

break;

}

}

fclose(fp);

}

void bookedCarDetail()

{

return\_car();

return\_car\_delete();

return\_car();

return\_car\_delete();

clrscr();

FILE \* fp = fopen("customer.bin","rb");

if(fp == NULL)

{

textcolor(LIGHTRED);

gotoxy(32,22);

printf("File cannot opened !!!");

return;

}

Customer\_Car\_Details CC;

int i;

textcolor(YELLOW);

gotoxy(31,1);

printf("CAR RENTAL SYSTEM\n");

for(i = 1; i <= 80 ; i++)

{

printf("%c",247);

}

gotoxy(31,5);

printf("\* BOOKED CAR DETAILS \*");

gotoxy(1,7);

textcolor(LIGHTGREEN);

for( i = 1; i<= 80 ; i++)

printf("%c",247);

gotoxy(1,8);

printf("Model");

gotoxy(14,8);

printf("Cust Name");

gotoxy(28,8);

printf("PickUp");

gotoxy(42,8);

printf("Drop");

gotoxy(56,8);

printf("Start\_Date");

gotoxy(68,8);

printf("End\_Date");

gotoxy(1,9);

for(i = 1; i<= 80 ; i++)

printf("%c",247);

int row = 10;

textcolor(YELLOW);

while(fread(&CC,sizeof(Customer\_Car\_Details),1,fp) == 1)

{

gotoxy(1,row);

printf("%s",getCarName(CC.car\_id));

gotoxy(14 ,row);

printf("%s",CC.cust\_name);

gotoxy(28,row);

printf("%s",CC.pick);

gotoxy(42,row);

printf("%s",CC.drop);

gotoxy(56,row);

printf("%d-%d-%d",CC.sd.tm\_mday,CC.sd.tm\_mon,CC.sd.tm\_year);

gotoxy(68,row);

printf("%d-%d-%d",CC.ed.tm\_mday,CC.ed.tm\_mon,CC.ed.tm\_year);

row++;

}

fclose(fp);

getch();

}

int checkDate(struct tm sd, struct tm ed)

{

int i,month\_days,current = 0,start = 0,end = 0;

int arr[] = {31,28,31,30,31,30,31,31,30,31,30,31};

time\_t now = time(0);

struct tm \*ptr\_now = localtime(&now);

current = ptr\_now->tm\_yday;

current += 1;

for(i = 0 ; i <= sd.tm\_mon-2 ;i++)

start += arr[i];

start += sd.tm\_mday;

for(i = 0 ; i <= ed.tm\_mon-2 ;i++)

end += arr[i];

end += ed.tm\_mday;

if((current <= start) && (current < end) && (start <= end))

return 1 ;

else

return 0;

}

void returnUpdateCarCount(int c\_id)

{

FILE \* fp = fopen("car.bin","rb+");

if(fp == NULL)

{

textcolor(LIGHTRED);

return ;

}

Car C;

while(fread(&C,sizeof(C),1,fp) == 1)

{

if(C.car\_id == c\_id)

{

int x = C.car\_count;

++x;

fseek(fp,-8,SEEK\_CUR);

fwrite(&x,sizeof(int),1,fp);

break;

}

}

fclose(fp);

}

void return\_car()

{

FILE \* fp = fopen("customer.bin","rb");

if(fp == NULL)

{

//message

}

int arr[] = {31,28,31,30,31,30,31,31,30,31,30,31};

time\_t now = time(0);

struct tm \*ptr\_now = localtime(&now);

struct tm temp;

int i;

int id;

int current\_day = ptr\_now->tm\_yday;

int current\_year = ptr\_now->tm\_year + 1900;

int end\_day;

Customer\_Car\_Details cc;

while( fread(&cc,sizeof(cc),1,fp) == 1)

{

end\_day = 0;

temp= cc.ed;

id = cc.car\_id;

for(i = 0 ; i <= temp.tm\_mon-2 ; i++)

end\_day += arr[i];

end\_day += temp.tm\_mday;

if((current\_day >= end\_day ) && (current\_year == temp.tm\_year))

{

returnUpdateCarCount(cc.car\_id);

}

else if(current\_year > temp.tm\_year)

{

returnUpdateCarCount(cc.car\_id);

}

}

fclose(fp);

}

void return\_car\_delete()

{

FILE \*fp1 = fopen("customer.bin","rb");

FILE \*fp2 = fopen("temp.bin","wb+");

Customer\_Car\_Details cc;

while(fread(&cc,sizeof(cc),1,fp1) == 1)

{

fwrite(&cc,sizeof(cc),1,fp2);

//printf("%d %s %s %s",cc.car\_id,cc.cust\_name,cc.pick,cc.drop);

//getch();

}

fclose(fp1);

fclose(fp2);

FILE \* temp1 = fopen("temp.bin","rb");

FILE \*temp2 = fopen("temp2.bin","wb");

int found = 0;

int result;

int arr[] = {31,28,31,30,31,30,31,31,30,31,30,31};

time\_t now = time(0);

struct tm \*ptr\_now = localtime(&now);

struct tm temp;

int i;

int id;

int current\_day = ptr\_now->tm\_yday;

int current\_year = ptr\_now->tm\_year + 1900;

int end\_day;

while( fread(&cc,sizeof(cc),1,temp1) == 1)

{

end\_day = 0;

temp= cc.ed;

id = cc.car\_id;

for(i = 0 ; i <= temp.tm\_mon-2 ; i++)

end\_day += arr[i];

end\_day += temp.tm\_mday;

if(((current\_day >= end\_day ) && (current\_year == temp.tm\_year)) ||(current\_year > temp.tm\_year))

{

found = 1;

//printf("\nfound\n");

//getch();

}

else

{

fwrite(&cc,sizeof(cc),1,temp2);

//printf("\nNot Match Date\n");

//getch();

}

}

fclose(temp1);

fclose(temp2);

if(found == 0)

{

remove("temp2.bin");

remove("temp.bin");

}

else{

result = remove("temp1.bin");

remove("customer.bin");

if(result != 0 )

{

//printf("\nremove\n");

//getch();

}

result = rename("temp2.bin","customer.bin");

if(result != 0)

{

//printf("\nremove\n");

//getch();

}

}

}