```cpp
#include <fstream.h>
#include <iomanip.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
#include <ctype.h>
#include "design.h"
char op;
void menu();
void delay() {
  for (unsigned int i = 0; i < 5000; i++)
    for (unsigned int j = 0; j < 9000; j++) { /*To delay output*/
    }
}
void loading(short x, short y, short N = 7)
/* (x,y)-Position of Load Animation
        Loading Running delay(N=1 for 0.4786 s delay approx.)
*/
{
  const char slash[5] = {92, 45, 47, 124};
  for (unsigned short i = 0; i < N; i++)
    for (short j = 0; j < 4; j++) {
      gotoxy(x, y);
      cout << slash[j];
      delay();
      delay();
    }
}
void clear(short x, short y, short n, short l = 1)
/* (x,y)-margin; n - no. of characters to clear
        l - no. of lines to clear
*/
{
  for (short i = 0; i < l; i++)
    for (short j = 0; j < n; j++) {
      gotoxy(x + j, y + i);
      cout << " ";
    }
}
void line(int p, int q, int l, char type = 'v', int type2 = 0) {
  int i;
  char hl, vl;
  /* Where (p,q)-margin (coordinate of top-left vertice of box border);
          type=v - For Vertical Line (default)
          type=h - For Horizontal Line
          l-length(horizontal distance) of border;
  */
  if (type2 == 1) {
    hl = '*';
    vl = '!';
  } else {
    hl = 45;
    vl = 124;
  }  // hyphen(-)and or (|)
  //(WHITE);
  if (type == 'v')
    for (i = 0; i < l; i++)  // Vertical line of length l
    {
      gotoxy(p, q + i);
      printf(
          "%c",
          vl);  //'7C'(hyphen); 'DE'(accented i); '7C' (or line); 'B3'
(border);
    }
  else
    for (i = 0; i < l; i++)  // Horizontal Line of length l
    {
      gotoxy(p + i, q);
      printf("%c", hl);
    }
}
void border(short p, short q, short l, short w,
            short type = 0)  // Use to make borders as the name suggests
{ /* Where (p,q)-margin (coordinate of top-left vertice of box border);
               w-width(vertical distance) of border;
               l-length(horizontal distance) of border;
               type => 2(double border); type => 0,1(single border); type =>
     3(slowmotion)
       */
  char cc, hl, vl;
  if (type == 2) {
    cc = '/';
    hl = '/';
    vl = '/';
  } else if (type == 1) {
    cc = '*';
    hl = '+';
    vl = '!';
  } else if (type == 4 || type == 0) {
    cc = '+';
    hl = '*';
    vl = '!';
  } else {
    cc = '*';
    hl = '*';
    vl = '*';
  }
  gotoxy(p, q);
  printf("%c", cc);  // top-left
  gotoxy(p + l, q);
  printf("%c", cc);  // top-right
  gotoxy(p, q + w);
  printf("%c", cc);  // bottom-left
  gotoxy(p + l, q + w);
  printf("%c", cc);  // bottom-right
  for (int i = 1, j = 1, m = (w - 1), n = (l - 1);; i++, j++, m--, n--) {
    if (i < w) {
      gotoxy(p, q + i);
      if (type == 2) cout << vl;
      printf("%c", vl);
    }
    if (j < l) {
      gotoxy(p + i, q);
      printf("%c", hl);
      if (type == 2) cout << hl;
    }
    if (type == 4) {
      delay();
    }  // for delay
    if (m > 0) {
      gotoxy(p + l, q + m);
      if (type == 2) cout << '\b' << vl;
      printf("%c", vl);
    }
    if (n > 0) {
      gotoxy(p + n, q + w);
      printf("%c", hl);
      if (type == 2) cout << hl;
    }
    if (type == 4) {
      delay();
    }  // for delay
    if (j >= l && i >= w) break;
  }
}
void cursor()  // To get position of pointer in console window(for output
               // purpose)
{
  int x = 1, y = 1;
  char key;
  do {
    gotoxy(x, y);
    cout << "      ";
    gotoxy(x, y);
    key = getch();
    if (key == ' ')
      x++;
    else if (key == '\b')
      x--;
    else if (key == '\r')
      y++;
    else if (key == 'p') {
      gotoxy(1, 1);
      cout << x << " " << y;
    } else if (key == 'r') {
      x = 1;
      y = 1;
    }
  } while (key != 'N');
}
void ventry(char t[], short code = 4, short N = 1)
```

```
/* code=0 >> for inputing alphabets(a-Z), space, comma(,) & dot(.)
        code=1 >> for inputing numbers(0-9), dot(.) & '/'
        code=2 >> for inputing username / ID
        code=3 >> for normal input without space, ':' & ';'
        code=4 >> for menu input (only numbers 0-9)
*/
{
  unsigned short i = 0;
  if (code == 0) {
    while (i < N && (t[i] = getch()) != '\r')
      if ((t[i] >= 97 && t[i] <= 122) || (t[i] >= 65 && t[i] <= 90) ||
          t[i] == 32 || t[i] == '.' || t[i] == ',') {
        if (t[0] != ' ' &&
            t[0] != '_')  // To prevent space or '_' at first input
        {
          cout << t[i];
          i++;
        }
      } else if (t[i] == 8 && i > 0) {
        cout << "\b \b";
        i--;
      }
  } else if (code == 1) {
    while (i < N && (t[i] = getch()) != '\r')
      if ((t[i] >= 48 && t[i] <= 57) || t[i] == 46 || t[i] == '/') {
        cout << t[i];
        i++;
      } else if (t[i] == 8 && i > 0) {
        cout << "\b \b";
        i--;
      }
  }
  if (code == 2) {
    while (i < N && (t[i] = getch()) != '\r')
      if ((t[i] >= 95 && t[i] <= 122) || (t[i] >= 46 && t[i] <= 57) ||
          (t[i] >= 65 && t[i] <= 90) || t[i] == '@') {
        cout << t[i];
        i++;
      } else if (t[i] == 8 && i > 0) {
        cout << "\b \b";
        i--;
      }
  }
  if (code == 3) {
    while (i < N && (t[i] = getch()) != '\r')
      if ((t[i] >= 33 && t[i] <= 126) && t[i] != ':' && t[i] != ';') {
        cout << t[i];
        i++;
      } else if (t[i] == 8 && i > 0) {
        cout << "\b \b";
        i--;
      }
  } else if (code == 4) {
    while (i < N && (t[i] = getch()) != '\r')
```

```
      if (t[i] >= 48 &&
          t[i] <= 57) {  // cout<<t[i]; //Uncomment to display input no.
        i++;
      } else if (t[i] == 8 && i > 0) {
        cout << "\b \b";
        i--;
      }
  }
  t[i] = '\0';
}
void voutput(short N, char t[], char o[] = NULL)
/* N >> for outputing N characters of the concatinated string
*/
{
  unsigned short i = 0;
  strcat(t, " ");
  strcat(t, o);
  if (N < strlen(t)) {
    for (i = N; i < N + 3; i++) t[i] = '.';
    strcat(t, "..");
    N += 2;
  }
  for (i = 0; i < N && i < strlen(t); i++) cout << t[i];
}
void About() {
  border(3, 2, 74, 22, 2);
  gotoxy(17, 4);
  cout << "A B O U T   I N F I N I T Y   C A R   R E N T A L";
  gotoxy(17, 5);
  cout << "--------------------------------------------------";
  gotoxy(11, 6);
  cout << "Our company started in 2001 in a small town in Joerdhailun";
  gotoxy(14, 8);
  cout << "which has now expanded globally having locations in";
  gotoxy(12, 10);
  cout << "Asia, Europe, North America. We have received numerous";
  gotoxy(12, 12);
  cout << "accolades for our service and the quality of our cars.";
  gotoxy(15, 14);
  cout << "We strive to provide our clients with quality our";
  gotoxy(17, 16);
  cout << "clients with quality cars at affordable prices.";
  gotoxy(13, 18);
  cout << "We have a wide selection of cars including high range";
  gotoxy(18, 20);
  cout << "of luxury cars like Rolls Royce, Bentley";
  gotoxy(22, 22);
  cout << "and even a few exotic sports cars. ";
}
class Login {
  char usr[13];
  char pss[13];

 public:
  int check();
} L;
int Login::check() {
xx:
  clrscr();
  border(3, 2, 74, 21, 2);
  int i = 0;
  gotoxy(23, 4);
  printf("W E L C O M E   T O  .  .  .");
  gotoxy(23, 5);
  cout << "-------------------------------------";
  gotoxy(23, 6);
  cout << "I N F I N I T Y   C A R   R E N T A L";
  gotoxy(23, 7);
  printf("-------------------------------------");
  gotoxy(34, 10);
  printf("SYSTEM LOGIN");
  gotoxy(34, 11);
  printf("------------");
  border(25, 12, 29, 7, 1);
  gotoxy(28, 14);
  cout << "Username : ";
  gotoxy(28, 17);
  cout << "Password : ";
  gotoxy(39, 14);
  ventry(usr, 2, 13);
  gotoxy(39, 17);
  while (i < 12 && (pss[i] = getch()) != '\r') {
    if (pss[i] == '\b' && i > 0) {
      cout << '\b' << ' ' << '\b';
      i--;
    } else if (i != 0 || pss[i] != '\b') {
      cout << "*";
      i++;
    }
  }
  pss[i] = '\0';
  if (strcmp(usr, "admin") == 0 && strcmp(pss, "pass123") == 0) {
    cout << '\a';
    clrscr();
    return (1);
  } else {
    gotoxy(24, 21);
    printf("Invalid Username Or Password !!!");
    getch();
    goto xx;
  }
}
short namecmp(char *name, char *word)
// To search for a 'word' in a 'name'
{
  if (strlen(name) < strlen(word)) {
    clrscr();
    cout << "Error exit function! String size less than word size.... ";
    return (-1);
  }
  short sc, count = 0;
  for (short i = 0; i <= (strlen(name) - strlen(word)); i++) {
    sc = 0;
    for (short j = 0; j < strlen(word); j++)
      if (tolower(name[i + j]) != tolower(word[j])) {
        sc = 1;
        break;
      }
    if (sc == 0) count++;
  }
  return count;
}
//****************################## class Cars
//****************##################
//## stored in cardata
class Cars {
  char carid[6];
  char make[20];
  char model[20];
  char year[5];
  char category[20];
  char passengers;
  float price;
  short status;  // 1=avail. , 0=booked
 public:
  void getcar();
  void putcar();
  void makecarid();
  void CarTable();
  void AvailTable(char *);
  void switch_status(int op = 2);
  char *retcarid() { return (carid); }
  char *retmodel() { return (model); }
  char *retmake() { return (make); }
  float retprice() { return (price); }
  ~Cars() { /*Destructor to distroy this class when the scope is over*/
  }
} C;
void Cars::getcar() {
  cout << "\n\tMAKE => ";
  gets(make);
  cout << "\n\tModel => ";
  gets(model);
  cout << "\n\tYEAR => ";
  ventry(year, 1, 5);
  cout << "\n\tCATEGORY => ";
  ventry(category, 0, 16);
  cout << "\n\tPassengers => ";
  cin >> passengers;
  cout << "\n\tPrice per day of car => ";
  cin >> price;
  status = 1;  // status=1 means available;
```

```cpp
  makecarid();
  cout << "\n\tCarID => " << carid;
  getch();
}
void Cars::putcar() {
  cout << "\n\tCarID : " << carid;  // Displayed on Table
  cout << "\n\n\tMAKE : " << make;  // MAKE+model Displayed on table
  cout << "\t\tModel : " << model;
  cout << "\n\n\tYEAR : " << year;
  cout << "\t\tCATEGORY : " << category;
  cout << "\n\n\tPassengers : " << passengers;
}
void Cars::makecarid() {
  short lm = strlen(make);
  short ly = strlen(year);
  int pos = 0, i, j;
  for (j = 0; j < 20; j++) i = random(4);
  if (i == 0) {
    carid[pos++] = tolower(make[0]);
    carid[pos++] = tolower(make[1]);
  }
  if (i == 1) {
    carid[pos++] = make[0];
    carid[pos++] = make[lm - 1];
  }
  if (i == 2) {
    carid[pos++] = make[0];
    carid[pos++] = make[1];
    carid[pos++] = make[2];
  }
  if (i == 3) {
    carid[pos++] = year[ly - 2];
    carid[pos++] = year[ly - 1];
    carid[pos++] = model[0];
    carid[pos++] = model[2];
  }
  randomize();
  for (j = 0; j < 20; j++) i = random(4);
  if (i == 0) {
    carid[pos++] = year[ly - 2];
    carid[pos++] = year[ly - 1];
  }
  if (i == 1) {
    carid[pos++] = model[0];
    carid[pos++] = model[1];
  }
  if (i == 2 && pos < 3) {
    carid[pos++] = model[0];
    carid[pos++] = model[1];
    carid[pos++] = model[2];
  }
  if (i == 3) {
    carid[pos++] = tolower(make[0]);
    carid[pos++] = '#';
  }
  for (i = 0; i < 5; i++)
    if (carid[i] == NULL || carid[i] == ' ') carid[i] = passengers;
  carid[5] = '\0';
}
void Cars::CarTable() {
  border(4, 3, 73, 20);
  line(8, 4, 19);  // Vertical line After No.
  line(16, 4, 19);  // Vertical line After Carid
  line(37, 4, 19);  // Vertical line After Car
  line(55, 4, 19);      // After Category
  line(65, 4, 19);      // After Price
  line(5, 5, 72, 'h');  // Horizontal line
  gotoxy(5, 4);
  cout << " No";
  gotoxy(9, 4);
  cout << " CarID";
  gotoxy(25, 4);
  cout << "CAR";
  gotoxy(43, 4);
  cout << "CATEGORY";
  gotoxy(56, 4);
  cout << "Price/day";
  gotoxy(69, 4);
  cout << "STATUS";
  gotoxy(1, 1);
  int i = 0, r = 7;
  ifstream carfile("carsdata", ios::in | ios::binary);
  gotoxy(20, 2);
  cout << "A L L   C A R S   R E P O R T   T A B L E";
  // Read From File
  while (carfile.read((char *)&C, sizeof(Cars))) {
    gotoxy(6, r);
    (i > 8) ? cout << (i + 1) : cout << " " << (i + 1);
    gotoxy(10, r);
    cout << carid;
    gotoxy(18, r);
    voutput(17, make, model);
    gotoxy(39, r);
    cout << category;
    gotoxy(59, r);
    cout << C.price;
    gotoxy(67, r);
    (status == 1) ? cout << "Available" : cout << "BOOKED!";
    i++;
    r++;
  }
}
void Cars::AvailTable(char *rentcarid) {
  clrscr();
  border(4, 4, 73, 17);
  line(9, 5, 16, 'v');  // Vertical line After No.
  line(16, 5, 16, 'v');  // Vertical line After Carid
  line(39, 5, 16, 'v');  // Vertical line After Car
  line(58, 5, 16, 'v');  // After Category
  line(70, 5, 16, 'v');  // After Price
  line(5, 6, 72, 'h');   // Horizontal line
  gotoxy(5, 5);
  cout << " No";
  gotoxy(10, 5);
  cout << " Year";
  gotoxy(26, 5);
  cout << "CAR";
  gotoxy(45, 5);
  cout << "CATEGORY";
  gotoxy(60, 5);
  cout << "Price/day";
  gotoxy(71, 5);
  cout << "Seats";
  gotoxy(1, 2);
  ifstream carfile("carsdata", ios::in | ios::binary);
  // Read From File
  int i, r;
  cout << "\t\t\tDISPLAY AVAILABLE CARS";
  for (i = 1, r = 8; carfile.read((char *)&C, sizeof(C));)
    if (status == 1) {
      gotoxy(5, r);
      cout << "[" << i << "]";
      gotoxy(11, r);
      cout << year;
      gotoxy(18, r);
      voutput(19, make, model);
      gotoxy(41, r);
      cout << category;
      gotoxy(63, r);
      cout << C.price;
      gotoxy(73, r);
      cout << passengers;
      i++, r++;
    }
  carfile.clear();
  carfile.seekg(0);
  gotoxy(5, 24);
  cout << "Select Car No. >> ";
  cin >> r;
  i = 1;
  while (carfile.read((char *)&C, sizeof(C))) {
    if (i == r && status == 1) {
      strcpy(rentcarid, carid);
      break;
    }
    if (status == 1) {
      i++;
    }
  }
}
void Cars::switch_status(int op) {
  if (op == 1)
```

```
    status = 1;
  else if (op == 0)
    status = 0;
  else
    (status == 0) ? status = 1 : status = 0;
}
void status_edit(char *op = "#Nothing") {
  fstream carfile("carsdata", ios::in | ios::out | ios::binary);
  while (carfile.read((char *)&C, sizeof(C))) {
    if (*op == '1' && *(op + 1) == '\0')
      C.switch_status(1);
    else if (*op == '0' && *(op + 1) == '\0')
      C.switch_status(0);
    else if (strcmp(C.retcarid(), op) == 0) {
      C.switch_status();
      carfile.seekg(carfile.tellg() - sizeof(C));
      carfile.write((char *)&C, sizeof(C));
      break;
    }
    carfile.seekg(carfile.tellg() - sizeof(C));
    carfile.write((char *)&C, sizeof(C));
  }
}
void AddCar() {
  Cars C;
  int i, N;
  cout << "\nEnter no. of cars to be added => ";
  cin >> N;
  clrscr();
  ofstream carfile("carsdata", ios::out | ios::binary | ios::app);
  // Write TO File
  cout << "\n\t\t\tADD NEW CAR";
  for (i = 0; i < N; i++) {
    clrscr();
    cout << "\n\n\tAdd Car No. " << (i + 1) << endl;
    C.getcar();
    carfile.write((char *)&C, sizeof(Cars));
  }
  carfile.close();
}
short SearchCar(Cars &D, char *objective = NULL)
/* Can only search for make, model, carid and price
*/
{
  short count = 0, searchstatus = 0, i = 0;
  Cars M[10];
  ifstream carfile("carsdata", ios::in | ios::out | ios::binary);
  if (*objective == NULL) return 0;
  // Read from File and Write to File
  while (carfile.read((char *)&C, sizeof(Cars))) {
    if (strcmp(objective, C.retcarid()) == 0 ||
        strcmpi(objective, C.retmodel()) == 0) {
      D = C;
      searchstatus++;
      break;
    } else if (strcmpi(objective, C.retmake()) == 0)
      count++;
  }
  carfile.clear();
  carfile.seekg(0);
  if (count == 1)
    while (carfile.read((char *)&C, sizeof(Cars)))
      if (strcmpi(objective, C.retmake()) == 0) {
        D = C;
        searchstatus++;
        break;
      }
  if (count > 1) {  // clrscr();
    gotoxy(13, 13);
    cout << "More than one record found !\a";
    while (carfile.read((char *)&C, sizeof(Cars)))
      if (strcmpi(objective, C.retmake()) == 0) {
        M[i] = C;
        i++;
      }
    for (i = 0; i < count; i++) {
      gotoxy(9, 15 + i);
      cout << "[" << (i + 1) << "]"
           << "\tCarID: " << M[i].retcarid();
      gotoxy(36, 15 + i);
      cout << "CAR: " << M[i].retmake() << " " << M[i].retmodel();
    }
    int op;
    gotoxy(6, 18);
    cout << " => ";
    cin >> op;
    carfile.clear();
    D = M[op - 1];
    searchstatus++;
  }
  carfile.close();
  if (searchstatus)
    return 1;
  else
    return 0;
}
void UpdateCar() {
  short updatestatus = 0;
  Cars D;
  char update[20];
  fstream carfile("carsdata", ios::in | ios::out | ios::binary);
  // Read from File and Write to File
  clrscr();
  cout << "\nEnter Carid/Model/Make of Car to Update => ";
  gets(update);
  SearchCar(D, update);
  D.putcar();
  getch();
  while (carfile.read((char *)&C, sizeof(Cars)))
    if (strcmp(update, C.retcarid()) == 0 ||
        strcmpi(update, C.retmodel()) == 0 ||
        strcmp(D.retcarid(), C.retcarid()) == 0) {
      cout << "\nCar Found!";
      updatestatus = 1;
      carfile.seekg(carfile.tellg() - sizeof(Cars));
      C.putcar();
      C.getcar();
      carfile.write((char *)&C, sizeof(Cars));
      break;
    }
  if (updatestatus)
    cout << "\n\t\tCar Details Updated ! ";
  else
    cout << "\n\t\tCar Not Found !";
  carfile.close();
  getch();
}
void SearchCar_UI() {
  char info[30];
search:
  clrscr();
  border(3, 2, 74, 21, 2);
  gotoxy(30, 4);
  cout << "SEARCH AND DISPLAY CAR";
  gotoxy(30, 5);
  cout << "----------------------";
  gotoxy(8, 7);
  cout << "Carid: ";
  gotoxy(8, 9);
  cout << "Model: ";
  gotoxy(8, 11);
  cout << "MAKE: ";
  gotoxy(36, 7);
  cout << "# Press ENTER to skip";
  gotoxy(36, 8);
  cout << "# Atleast one field should be entered";
  gotoxy(36, 9);
  cout << "# Model=Car Model(eg. Corolla,Innova)";
  gotoxy(36, 10);
  cout << "# MAKE=Company(eg. BMW,TOYOTA)";
  line(6, 12, 45, 'h');
  gotoxy(15, 7);
  gets(info);
  if (info[0] == NULL || SearchCar(C, info) == 0) {
    gotoxy(15, 9);
    gets(info);
    if (info[0] == NULL || SearchCar(C, info) == 0) {
      gotoxy(14, 11);
      gets(info);
      if (SearchCar(C, info) == 0) {
        gotoxy(8, 13);
        cout << "RECORD NOT FOUND !";
        gotoxy(8, 15);
        cout << "Do you wish to continue (y/n)....";
        if (tolower(::op = getch()) == 'n') {
          flushall();
          return;
        }
        goto search;
      }
    }
  }
  flushall();
  clear(7, 13, 65, 7);
  gotoxy(13, 12);
  C.putcar();
  cout << "\n\n\t\tDo you wish to search again ? (y/n)......";
  border(3, 2, 74, 21, 2);
choose:
  if (tolower(::op = getch()) == 'y')
    goto search;
  else if (::op == 'n') {
    flushall();
    return;
  } else
    goto choose;
}
void DeleteCar() {
  char info[30];
search:
  clrscr();
  border(3, 2, 74, 21, 2);
  gotoxy(35, 4);
  cout << "DELETE CAR";
  gotoxy(35, 5);
  cout << "----------";
  gotoxy(8, 7);
  cout << "Carid: ";
  gotoxy(8, 9);
  cout << "Model: ";
  gotoxy(8, 11);
  cout << "MAKE: ";
  gotoxy(36, 7);
  cout << "# Press ENTER to skip";
  gotoxy(36, 8);
  cout << "# Atleast one field should be entered";
  gotoxy(36, 9);
  cout << "# Model=Car Model(eg. Corolla,Innova)";
  gotoxy(36, 10);
  cout << "# MAKE=Company(eg. BMW,TOYOTA)";
  line(6, 12, 45, 'h');
  gotoxy(15, 7);
  gets(info);
  if (info[0] == NULL || SearchCar(C, info) == 0) {
    gotoxy(15, 9);
    gets(info);
    if (info[0] == NULL || SearchCar(C, info) == 0) {
      gotoxy(14, 11);
      gets(info);
      if (SearchCar(C, info) == 0) {
        gotoxy(8, 13);
        cout << "RECORD NOT FOUND !";
        gotoxy(8, 15);
        cout << "Do you wish to continue (y/n)....";
        if (tolower(::op = getch()) == 'n') {
          flushall();
          return;
        }
        goto search;
      }
    }
  }
  flushall();
  clear(7, 13, 65, 7);
  gotoxy(13, 12);
  C.putcar();
  border(3, 2, 74, 21, 2);
  gotoxy(17, 21);
  cout << '_';
  gotoxy(17, 22);
  cout << "Delete";
  gotoxy(38, 21);
  cout << '_';
  gotoxy(38, 22);
  cout << "Retry";
  gotoxy(58, 21);
  cout << '_';
  gotoxy(58, 22);
  cout << "Cancel";
choose:
  if (tolower(::op = getch()) == 'r')
    goto search;
  else if (::op == 'c') {
    flushall();
    return;
  } else if (::op == 'd') {
  del:
    clrscr();
    border(3, 2, 74, 21, 2);
    gotoxy(8, 4);
    cout << "\aAre you sure you want to DELETE RECORD !! (Y/N): ";
    ventry(&::op, 3, 2);
    if (::op == 'Y') {
      Cars D;
      ifstream carfile;
      ofstream temp;
      carfile.open("carsdata", ios::binary);
      temp.open("temp", ios::binary);
      while (carfile.read((char *)&D, sizeof(Cars)))
        if (strcmp(C.retcarid(), D.retcarid()) != 0)
```

```cpp
          temp.write((char *)&D, sizeof(D));
      temp.close();
      carfile.close();
      flushall();
      remove("carsdata");
      rename("temp", "carsdata");
      gotoxy(8, 6);
      cout << "RECORD Successfully DELETED....";
      getch();
    } else if (::op == 'N')
      return;
    else
      goto del;
  } else
    goto choose;
}
void DisplayCar() {
  Cars D;
  int i = 0;
  ifstream carfile("carsdata", ios::in | ios::binary);
  // Read From File
  clrscr();
  cout << "\n\t\t\tDISPLAY AVAILABLE CARS";
  while (carfile.read((char *)&D, sizeof(Cars))) {
    cout << "\n\n\tCar No. " << (i + 1) << endl;
    D.putcar();
    getch();
    i++;
  }
  carfile.close();
}
void rentcar_design() {
  border(3, 2, 74, 21, 2);
  gotoxy(9, 7);
  cout << "Drv. Lic. #: ";
  gotoxy(46, 7);
  cout << "Name: ";
  line(8, 8, 60, 'h');
  gotoxy(9, 10);
  cout << "PhoneNo. & or  Address: ";
  gotoxy(9, 12);
  cout << "City: ";
  gotoxy(46, 12);
  cout << "ZIPCode: ";
  gotoxy(9, 14);
  cout << "Email: ";
  gotoxy(46, 14);
  cout << "Rent Car ID: ";
  line(8, 16, 60, 'h');
  gotoxy(9, 17);
  cout << "FROM DATE (dd/mm/yyyy): ";
  gotoxy(45, 17);
  cout << "TO DATE (dd/mm/yyyy): ";
}
```

```
//############************* CLASS rentDate ***************##############
// TO Calculate difference in dates
struct date {
  unsigned short day, month, year;
};
void getdate(date &, short, short);
void putdate(date &, short, short);
class rentDate {
 public:
  date from, to;
  unsigned int convert_datetodays(date D, unsigned short y = 0);
  unsigned int calc_diff() {
    return (convert_datetodays(to, from.year) - convert_datetodays(from));
  }
  short leap_year(unsigned short year) {
    if ((year % 4 == 0) && !(year % 100 == 0) || year % 400 == 0)
      return 1;
    else
      return 0;
  }
  short verifydate(short, short code = 1);
  void Get_fromdate(short p, short q) { getdate(from, p, q); }
  void Get_todate(short p, short q) { getdate(to, p, q); }
  void Put_fromdate(short p, short q) { putdate(from, p, q); }
  void Put_todate(short p, short q) { putdate(to, p, q); }
  ~rentDate() { /*Destructor, the destroyer*/
  }
} A;
void getdate(date &d, short p, short q) {
in:
  clear(p, q, 12);
  gotoxy(p, q);
  cout << "  /  /  ";
  gotoxy(p, q);
  cin >> d.day;
  if (!A.verifydate(d.day, 1)) {
    cin.clear();
    cin.ignore(1000, '\n');
    goto in;
  }
  if (d.day < 10) {
    gotoxy(p, q);
    cout << 0 << d.day;
  }
  gotoxy(p + 3, q);
  cin >> d.month;
  if (!A.verifydate(d.month, 2)) {
    cin.clear();
    cin.ignore(100, '\n');
    goto in;
  }
  if (d.month < 10) {
    gotoxy(p + 3, q);
    cout << 0 << d.month;
  }
  gotoxy(p + 6, q);
  cin >> d.year;
  if (!A.verifydate(d.year, 3)) {
    cin.clear();
    cin.ignore(100, '\n');
    goto in;
  }
}
void putdate(date &d, short p, short q) {
  gotoxy(p, q);
  cout << setw(2) << setfill('0') << d.day << '/';
  cout << setw(2) << setfill('0') << d.month << '/' << d.year;
}
short rentDate::verifydate(short d, short code) {
  if (code == 2) {
    if (d >= 1 && d <= 12)
      return 1;
    else
      return 0;
  } else if (code == 3) {
    if (d >= 2016 && d <= 2070)
      return 1;
    else
      return 0;
  } else {
    if (d >= 1 && d <= 31)
      return 1;
    else
      return 0;
  }
}
unsigned int rentDate::convert_datetodays(date D, unsigned short y) {
  unsigned int days = 0;
  short i;
  for (unsigned short k = y; k < D.year; k++) {
    if (y == 0) break;
    if (leap_year(k))
      days += 366;
    else
      days += 365;
  }
  for (i = 1; i < D.month; i++) {
    if (i == 4 || i == 6 || i == 9 || i == 11)
      days += 30;
    else if (i == 2) {
      if (leap_year(D.year))
        days += 29;
      else
        days += 28;
    } else
      days += 31;
  }
  days += D.day;
```

```cpp
  return days;
}
//************/**************/************* class Customer
//*************/***************/*************/*****
/* stored in CUST_record */
class Customer {
  char lic_no[12];
  char name[30], email[30];
  char address[100], city[30];
  char rentcarid[10];
  char zipcode[10];

 public:
  rentDate T;
  void getcustomer();
  void putcustomer();
  void CustomerCar_table();
  void get_rentcarid() { C.AvailTable(rentcarid); }
  void customer_report();
  char *ret_rentcarid() { return rentcarid; }
  char *ret_licno() { return lic_no; }
  char *ret_name() { return name; }
  char *ret_email() { return email; }
  char *ret_address() { return address; }
  ~Customer() { /*Destructor the destroyer*/
  }
} N;
void Customer::getcustomer() {
lic:
  gotoxy(22, 7);
  ventry(lic_no, 1, 10);
  if (lic_no[0] == NULL) goto lic;
name:
  gotoxy(52, 7);
  ventry(name, 0, 23);
  if (name[0] == NULL) goto name;
  gotoxy(33, 10);
  gets(address);
  gotoxy(15, 12);
  gets(city);
  gotoxy(55, 12);
  ventry(zipcode, 1, 6);
  gotoxy(16, 14);
  ventry(email, 3, 26);
date:
  T.Get_fromdate(13, 18);
  T.Get_todate(49, 18);
  if (T.calc_diff() > 365) {
    gotoxy(13, 20);
    cout << "Sorry, maximum rent period is 365 days !.........";
    getch();
    clear(13, 18, 60, 4);
    goto date;
  }
  flushall();
}
void Customer::putcustomer() {
  gotoxy(22, 7);
  cout << lic_no;
  gotoxy(52, 7);
  cout << name;
  gotoxy(33, 10);
  cout << address;
  gotoxy(15, 12);
  cout << city;
  gotoxy(55, 12);
  puts(zipcode);
  gotoxy(16, 14);
  puts(email);
  gotoxy(59, 14);
  puts(rentcarid);
  T.Put_fromdate(13, 18);
  T.Put_todate(50, 18);
}
void Customer::CustomerCar_table() {
  clrscr();
  gotoxy(20, 2);
  cout << "C U S T O M E R - C A R  R E P O R T  T A B L E";
  ifstream customer("CUST_record", ios::in | ios::binary);
  if (!customer) {
    cout << "\n\n\t\aFile Does Not Exist / Unable to Open File ";
    getch();
  } else {
    Cars D;
    border(4, 3, 73, 20, 1);
    line(16, 4, 19);  // Vertical line After Lic. #
    line(38, 4, 19);  // Vertical line After Customer
    line(46, 4, 19);      // After CarID
    line(67, 4, 19);      // After CAR
    line(5, 5, 72, 'h');  // Horizontal line
    gotoxy(5, 4);
    cout << " Licence #";
    gotoxy(23, 4);
    cout << " CUSTOMER ";
    gotoxy(39, 4);
    cout << " CarID";
    gotoxy(54, 4);
    cout << " CAR";
    gotoxy(68, 4);
    cout << "Price/day";
    short i = 0, r = 7;
    // Read From File
    while (customer.read((char *)&N, sizeof(N))) {
      SearchCar(D, rentcarid);
      gotoxy(40, r);
      cout << rentcarid;
      gotoxy(6, r);
      cout << lic_no;
      gotoxy(48, r);
      voutput(17, D.retmake(), D.retmodel());
      gotoxy(18, r);
      cout << name;
      gotoxy(71, r);
      cout << D.retprice();
      // getch();
      i++;
      r += 2;
    }
  }
}
void Customer::customer_report() {
  clrscr();
  gotoxy(22, 2);
  cout << "A L L   C U S T O M E R   R E P O R T";
  ifstream customer("CUST_record", ios::in | ios::binary);
  if (!customer) {
    gotoxy(8, 4);
    cout << "\aFile Does Not Exist / Unable to Open File ";
  } else {
    Cars D;
    border(4, 3, 73, 20, 1);
    gotoxy(2, 24);
    cout << "* Price may vary depending on the rented car and returning day";
    gotoxy(4, 25);
    cout << "and offers included";
    line(16, 4, 19);  // Vertical line After Lic. #
    line(38, 4, 19);  // Vertical line After Customer
    line(49, 4, 19);      // After fromdate
    line(60, 4, 19);      // After todate
    line(67, 4, 19);      // After days
    line(5, 5, 72, 'h');  // Horizontal line
    gotoxy(5, 4);
    cout << " Licence #";
    gotoxy(23, 4);
    cout << " CUSTOMER ";
    gotoxy(40, 4);
    cout << "From Date";
    gotoxy(51, 4);
    cout << "To Date";
    gotoxy(61, 4);
    cout << " Days";
    gotoxy(69, 4);
    cout << "*Price";
    short i = 0, r = 7;
    // Read From File
    while (customer.read((char *)this, sizeof(Customer))) {
      SearchCar(D, rentcarid);
      gotoxy(6, r);
      cout << lic_no;
      gotoxy(18, r);
      cout << name;
      T.Put_fromdate(39, r);
```

```cpp
      T.Put_todate(50, r);
      gotoxy(62, r);
      cout << T.calc_diff();
      gotoxy(69, r);
      cout << (T.calc_diff() * D.retprice());
      i++;
      r += 2;  // getch();
    }
  }
}
void AddCustomer() {
  clrscr();
  gotoxy(32, 4);
  cout << "RENT A CAR SYSTEM";
  gotoxy(32, 5);
  cout << "------------------";
  rentcar_design();
  N.getcustomer();
  N.get_rentcarid();
  clrscr();
  rentcar_design();
  gotoxy(32, 4);
  cout << "RENT A CAR SYSTEM";
  gotoxy(32, 5);
  cout << "------------------";
  N.putcustomer();
  gotoxy(23, 21);
  cout << '_';
  gotoxy(23, 22);
  cout << "Save";
  gotoxy(51, 21);
  cout << '_';
  gotoxy(51, 22);
  cout << "Cancel";
save:
  ::op = tolower(getch());
  if (::op == 's') {  // Write TO File
    ofstream cust;
    cust.open("CUST_record", ios::out | ios::binary | ios::app);
    cust.write((char *)&N, sizeof(class Customer));
    cust.close();
    flushall();
    status_edit(N.ret_rentcarid());
    clrscr();
    gotoxy(30, 12);
    cout << " PLEASE WAIT ! ";
    loading(52, 12);
    gotoxy(30, 12);
    cout << " CUSTOMER SAVED ! ";
    for (short i = 0; i < 5; i++) delay();
  } else if (::op == 'c') {
    flushall();
    return;
  } else
```

```
    goto save;
}
void DisplayCust() {
  clrscr();
  gotoxy(32, 4);
  cout << "RENT A CAR SYSTEM";
  gotoxy(32, 5);
  cout << "------------------";
  ifstream cust;
  cust.open("CUST_record", ios::in | ios::binary);
  cust.clear();
  cust.seekg(0);
  // Read From File
  while (cust.read((char *)&N, sizeof(class Customer))) {
    clear(4, 6, 73, 18);
    rentcar_design();
    N.putcustomer();
    flushall();
    getch();
  }
  cust.close();
}
short SearchCust(Customer &D, char *objective = "#Nothing", short name = 0)
/*  Can only search for Drv_Lic.#, *Name, email_id and rentcarid
         * more than one record;
*/ {
  short count = 0, searchstatus = 0, i = 0;
  Customer M[10];
  ifstream customer("CUST_record", ios::in | ios::out | ios::binary);
  if (*objective == NULL) return 0;
  // Read from File and Write to File
  while (customer.read((char *)&N, sizeof(Customer))) {
    if (strcmp(objective, N.ret_rentcarid()) == 0 ||
        strcmp(objective, N.ret_email()) == 0 ||
        strcmp(objective, N.ret_licno()) == 0) {
      D = N;
      searchstatus++;
      break;
    } else if (name == 1 && namecmp(N.ret_name(), objective))
      count++;
  }
  // gotoxy(1,1);cout<<"{ "<<count<<" }"; //For Testing
  customer.clear();
  customer.seekg(0);
  if (count == 1)
    while (customer.read((char *)&N, sizeof(Customer)))
      if (namecmp(N.ret_name(), objective)) {
        D = N;
        searchstatus++;
        break;
      }
  if (count > 1) {
    gotoxy(8, 16);
    cout << "More than one record found !\a";
```

```cpp
    while (customer.read((char *)&N, sizeof(Customer)))
      if (namecmp(N.ret_name(), objective)) {
        M[i] = N;
        i++;
      }
    gotoxy(9, 17);
    cout << "Option\t\t Drv_Lic.#"
         << "\t\t\tFull Name";
    gotoxy(7, 18);
    cout << "------------------------------------------------------------";
    for (i = 0; i < count; i++) {
      gotoxy(10, 19 + i);
      cout << "[" << (i + 1) << "]"
           << "\t\t" << setw(10) << setfill(' ') << M[i].ret_licno();
      cout << "\t\t" << setw(17) << M[i].ret_name();
    }
    int op;
  opt:
    gotoxy(8, 22);
    cout << "Enter Option / 0 to exit => ";
    cin >> op;
    if (op < i + 2) {
      if (op == 0) {
        clear(6, 16, 65, 7);
        return 0;
      }
      D = M[op - 1];
      searchstatus++;
    } else
      goto opt;
  }
  customer.close();
  if (searchstatus)
    return 1;
  else
    return 0;
}
void SearchCust_UI() {
  char info[30];
search:
  clrscr();
  border(3, 2, 74, 21, 2);
  gotoxy(26, 4);
  cout << "SEARCH AND DISPLAY CUSTOMER";
  gotoxy(26, 5);
  cout << "---------------------------";
  gotoxy(8, 8);
  cout << "Drv. Lic. #: ";
  gotoxy(8, 10);
  cout << "Rent Car ID: ";
  gotoxy(8, 12);
  cout << "Name: ";
  gotoxy(8, 14);
  cout << "Email: ";
  gotoxy(37, 8);
  cout << "# Press ENTER to skip";
  gotoxy(37, 10);
  cout << "# Atleast one field should be entered";
  gotoxy(37, 12);
  cout << "# Drv. Lic. # = Licence No.of Customer";
  gotoxy(37, 14);
  cout << "# Rent Car ID = CarID of rented Car";
  line(6, 15, 45, 'h');
  gotoxy(21, 8);
  gets(info);  // input lic_no.
  if (info[0] == NULL || SearchCust(N, info) == 0)  // search lic_no.
  {
    gotoxy(21, 10);
    gets(info);  // input rentcarid
    if (info[0] == NULL || SearchCust(N, info) == 0)  // search rentcarid
    {
      gotoxy(14, 12);
      gets(info);  // input name
      if (info[0] == NULL || SearchCust(N, info, 1) == 0)  // search name
      {
        gotoxy(15, 14);
        gets(info);  // input email
        if (SearchCust(N, info) == 0)  // search email
        {
          gotoxy(8, 16);
          cout << "RECORD NOT FOUND !";
          gotoxy(8, 18);
          cout << "Do you wish to continue (y/n)....";
          if (tolower(::op = getch()) == 'n') {
            flushall();
            return;
          }
          goto search;
        }
      }
    }
  }
  flushall();
  clrscr();
  rentcar_design();
  N.putcustomer();
  gotoxy(29, 4);
  cout << "RENTAL INFORMATION";
  gotoxy(29, 5);
  cout << "------------------";
  gotoxy(20, 21);
  cout << "Do you wish to search again ? (y/n)......";
choose:
  if (tolower(::op = getch()) == 'y')
    goto search;
  else if (::op == 'n') {
    flushall();
    return;
  } else
    goto choose;
}
void deletecust(Customer D) {
  ifstream customer;
  ofstream temp;
  customer.open("CUST_record", ios::binary);
  temp.open("temp", ios::binary);
  while (customer.read((char *)&N, sizeof(Customer)))
    if (strcmp(D.ret_licno(), N.ret_licno()) != 0)
      temp.write((char *)&N, sizeof(Customer));
  temp.close();
  customer.close();
  flushall();
  remove("Cust_record");
  rename("temp", "Cust_record");
}
void DeleteCust() {
  char info[30];
search:
  clrscr();
  border(3, 2, 74, 21, 2);
  gotoxy(33, 4);
  cout << "DELETE CUSTOMER";
  gotoxy(33, 5);
  cout << "---------------";
  gotoxy(8, 8);
  cout << "Drv. Lic. #: ";
  gotoxy(8, 12);
  cout << "Name: ";
  gotoxy(8, 14);
  cout << "Email: ";
  gotoxy(8, 10);
  cout << "Rent Car ID: ";
  gotoxy(37, 8);
  cout << "# Press ENTER to skip";
  gotoxy(37, 10);
  cout << "# Atleast one field should be entered";
  gotoxy(37, 12);
  cout << "# Drv. Lic. # = Licence No.of Customer";
  gotoxy(37, 14);
  cout << "# Rent Car ID = CarID of rented Car";
  line(6, 15, 45, 'h');
  gotoxy(21, 8);
  gets(info);  // input lic_no.
  if (info[0] == NULL || SearchCust(N, info) == 0)  // search lic_no.
  {
    gotoxy(21, 10);
    gets(info);  // input rentcarid
    if (info[0] == NULL || SearchCust(N, info) == 0)  // search rentcarid
    {
      gotoxy(14, 12);
      gets(info);  // input name
      if (info[0] == NULL || SearchCust(N, info, 1) == 0)  // search name
      {
        gotoxy(15, 14);
        gets(info);  // input email
        if (SearchCust(N, info) == 0)  // search email
        {
          gotoxy(8, 16);
          cout << "RECORD NOT FOUND !";
          gotoxy(8, 18);
          cout << "Do you wish to continue (y/n)....";
          if (tolower(::op = getch()) == 'n') {
            flushall();
            return;
          }
          goto search;
        }
      }
    }
  }
  flushall();
  clrscr();
  rentcar_design();
  N.putcustomer();
  gotoxy(29, 4);
  cout << "RENTAL INFORMATION";
  gotoxy(29, 5);
  cout << "------------------";
  gotoxy(20, 21);
  gotoxy(17, 21);
  cout << '_';
  gotoxy(17, 22);
  cout << "Delete";
  gotoxy(38, 21);
  cout << '_';
  gotoxy(38, 22);
  cout << "Retry";
  gotoxy(58, 21);
  cout << '_';
  gotoxy(58, 22);
  cout << "Cancel";
choose:
  if (tolower(::op = getch()) == 'r')
    goto search;
  else if (::op == 'c') {
    flushall();
    return;
  } else if (::op == 'd') {
  del:
    clrscr();
    border(3, 2, 74, 21, 2);
    gotoxy(8, 4);
    cout << "\aAre you sure you want to DELETE RECORD !! (Y/N): ";
    ventry(&::op, 3, 2);
    if (::op == 'Y') {
      status_edit(N.ret_rentcarid());
```

```cpp
      deletecust(N);
      gotoxy(8, 6);
      cout << "RECORD Successfully DELETED....";
      getch();
    } else if (::op == 'N')
      return;
    else
      goto del;
  } else
    goto choose;
}
class Bill {  // Customer details
  char lic_no[12];
  char name[30];
  char contact[100];
  // Car details
  char carid[10];
  char car[40];
  float carprice;
  rentDate D;
  date currentdate;
  float totalprice;

 public:
  void makeBill(Customer);
  void getcar(char *);
  // void getdate(date&,short,short);
  void calc_totalprice();
  unsigned int calc_returndaydiff();
  void editdate();
  void test() {
    gotoxy(8, 1);
    cout << totalprice;
  }
  void editprice();
  void printbill();
  void savebill();  // Saves Bill in binaryfile"bills.dat"
};
void Bill::makeBill(Customer N) {
  border(3, 2, 74, 21, 2);
  gotoxy(10, 4);
  cout << "Enter Current Date: ";
  getdate(currentdate, 30, 4);
  strcpy(lic_no, N.ret_licno());
  strcpy(name, N.ret_name());
  strcpy(contact, N.ret_address());
  getcar(N.ret_rentcarid());
  D = N.T;
  calc_totalprice();
}
unsigned int Bill::calc_returndaydiff() {
  signed int days;
  unsigned int a = A.convert_datetodays(currentdate, 2016);
  unsigned int b = A.convert_datetodays(D.to, 2016);
  days = a - b;
  if (days > 2)
    return days;
  else
    return 0;
}
void Bill::editdate() {
  clear(32, 15, 12);
  getdate(D.to, 32, 15);
  calc_totalprice();
}
void Bill::editprice() {
  clear(10, 20, 60);
  gotoxy(10, 20);
  cout << "New Amount ( Rs " << totalprice << " )= Rs ";
  cin >> totalprice;
}
void Bill::getcar(char *rentcarid) {
  Cars T;
  SearchCar(T, rentcarid);
  strcpy(carid, T.retcarid());
  strcpy(car, T.retmake());
  strcat(car, " ");
  strcat(car, T.retmodel());
  carprice = T.retprice();
}
void Bill::calc_totalprice() {
  totalprice = D.calc_diff() * carprice;
  if (calc_returndaydiff())
    totalprice += (carprice * 1.5 * calc_returndaydiff());
  if (totalprice > 100000) {
    // gotoxy(38,20);cout<<"You have recieved 25% discount!!"<<endl;
    totalprice *= 0.75;
  } else if (totalprice > 50000) {
    // gotoxy(38,20);cout<<"You have recieved 20% discount!!"<<endl;
    totalprice *= 0.8;
  } else if (totalprice > 25000) {
    // gotoxy(38,20);cout<<"You have recieved 15% discount!!";
    totalprice *= 0.85;
  } else if (totalprice > 10000) {
    // gotoxy(38,20);cout<<"You have recieved 10% discount!!";
    totalprice *= 0.9;
  }
}
void Bill::printbill() {
  gotoxy(2, 1);
  cout << "Current Date: ";
  putdate(currentdate, 15, 1);
  border(3, 2, 74, 22, 2);
  border(8, 5, 64, 16, 0);
  gotoxy(27, 4);
  cout << "C U S T O M E R   B I L L ";
  gotoxy(10, 6);
  cout << "CUSTOMER DETAILS";
  line(9, 7, 30, 'h', 1);
  gotoxy(10, 8);
  cout << "Drv. Lic. #: " << lic_no;
  gotoxy(46, 8);
  cout << "NAME: " << name;
  gotoxy(10, 9);
  cout << "Contact INFO: " << contact;
  line(9, 10, 30, 'h', 1);
  gotoxy(10, 11);
  cout << "RENTAL INFO";
  line(9, 12, 30, 'h', 1);
  gotoxy(10, 13);
  cout << "RentedCar_ID: " << carid;
  gotoxy(46, 13);
  cout << "CAR: " << car;
  gotoxy(10, 14);
  cout << "FROM DATE (dd/mm/yyyy): ";
  D.Put_fromdate(34, 14);
  gotoxy(10, 15);
  cout << "TO DATE (dd/mm/yyyy): ";
  D.Put_todate(32, 15);
  line(9, 16, 30, 'h', 1);
  gotoxy(10, 17);
  cout << "PRICE";
  line(9, 18, 30, 'h', 1);
  gotoxy(10, 19);
  cout << "Price/day for Car = Rs " << carprice;
  gotoxy(41, 19);
  cout << "No. of Days Rented = " << D.calc_diff();
  if (calc_returndaydiff()) cout << "+" << calc_returndaydiff();
  if (calc_returndaydiff()) {
    gotoxy(41, 20);
    cout << "Fine(Late Return) = Rs "
         << (1.5 * calc_returndaydiff() * carprice);
  }
  gotoxy(10, 20);
  cout << "Net Amount = Rs. " << totalprice;
}
void Bill::savebill() {
  ofstream bill;
  bill.open("bills.dat", ios::out | ios::app);
  bill.write((char *)this, sizeof(Bill));
  bill.close();
}
void ReturnCar() {
  char info[30];
search:
  clrscr();
  border(3, 2, 74, 21, 2);
  gotoxy(24, 4);
  cout << "R E T U R N   A N D   B I L L I N G";
  gotoxy(24, 5);
  cout << "-----------------------------------";
  gotoxy(8, 8);
```

```
    cout << "Drv. Lic. #: ";
    gotoxy(8, 10);
    cout << "Rent Car ID: ";
    gotoxy(8, 12);
    cout << "Name: ";
    gotoxy(8, 14);
    cout << "Email: ";
    gotoxy(37, 8);
    cout << "# Press ENTER to skip";
    gotoxy(37, 10);
    cout << "# Atleast one field should be entered";
    gotoxy(37, 12);
    cout << "# Drv. Lic. # = Licence No.of Customer";
    gotoxy(37, 14);
    cout << "# Rent Car ID = CarID of rented Car";
    line(6, 15, 45, 'h');
    gotoxy(21, 8);
    gets(info);  // input lic_no.
    if (info[0] == NULL || SearchCust(N, info) == 0)  // search lic_no.
    {
      gotoxy(21, 10);
      gets(info);  // input rentcarid
      if (info[0] == NULL || SearchCust(N, info) == 0)  // search rentcarid
      {
        gotoxy(14, 12);
        gets(info);  // input name
        if (info[0] == NULL || SearchCust(N, info, 1) == 0)  // search name
        {
          gotoxy(15, 14);
          gets(info);  // input email
          if (SearchCust(N, info) == 0)  // search email
          {
            gotoxy(8, 16);
            cout << "RECORD NOT FOUND !";
            gotoxy(8, 18);
            cout << "Do you wish to continue (y/n)....";
            if (tolower(::op = getch()) == 'n') {
              flushall();
              return;
            }
            goto search;
          }
        }
      }
    }
    flushall();
    clrscr();
    rentcar_design();
    N.putcustomer();
    gotoxy(29, 4);
    cout << "RENTAL INFORMATION";
    gotoxy(29, 5);
    cout << "------------------";
    gotoxy(20, 21);
```

```
  gotoxy(17, 21);
  cout << "     _";
  gotoxy(17, 22);
  cout << "View Bill";
  gotoxy(38, 21);
  cout << '_';
  gotoxy(38, 22);
  cout << "Retry";
  gotoxy(58, 21);
  cout << '_';
  gotoxy(58, 22);
  cout << "Cancel";
choose:
  if (tolower(::op = getch()) == 'r')
    goto search;
  else if (::op == 'c') {
    flushall();
    return;
  } else if (::op == 'b') {
    clrscr();
    Bill B;
    B.makeBill(N);
  billing:
    clrscr();
    B.printbill();
    gotoxy(11, 22);
    cout << '_';
    gotoxy(11, 23);
    cout << "Return Car";
    gotoxy(30, 22);
    cout << "     _";
    gotoxy(30, 23);
    cout << "Edit Price";
    gotoxy(49, 22);
    cout << "     _";
    gotoxy(49, 23);
    cout << "Edit Date";
    gotoxy(66, 22);
    cout << " _";
    gotoxy(66, 23);
    cout << "Exit";
    if (tolower(::op = getch()) == 'r') {
      clrscr();
      border(3, 2, 74, 21, 2);
      gotoxy(8, 4);
      cout << "Confirm Return Car [y/n] > ";
      cin >> ::op;
      if (tolower(op) == 'y') {
        status_edit(N.ret_rentcarid());
        deletecust(N);
        B.savebill();
        gotoxy(8, 6);
        cout << "Car Returned And Bill Saved......";
        getch();
```

```cpp
      } else
        goto billing;
    } else if (::op == 'p') {
      B.editprice();
      B.test();
      getch();
      goto billing;
    } else if (::op == 'd') {
      B.editdate();
      goto billing;
    } else if (::op == 'x') {
      flushall();
      clrscr();
      return;
    } else {
      cout << '\a';
      goto billing;
    }
  } else {
    cout << '\a';
    goto choose;
  }
}
void report_menu() {
menu:
  clrscr();
  border(3, 2, 74, 21, 2);
  border(23, 4, 37, 17, 0);
  gotoxy(30, 6);
  cout << "R E P O R T S   M E N U";
  gotoxy(29, 7);
  cout << "-------------------------";
  gotoxy(27, 9);
  cout << " [1] View All Car Reports";
  gotoxy(27, 11);
  cout << " [2] View All Customer Report";
  gotoxy(27, 13);
  cout << " [3] View Customer - Car Report";
  gotoxy(27, 15);
  cout << " [4] GO BACK\n";
  gotoxy(1, 1);
  ventry(&::op);
  switch (::op) {
    case '1':
      clrscr();
      C.CarTable();
      getch();
      break;
    case '2':
      clrscr();
      N.customer_report();
      getch();
      break;
    case '3':
      clrscr();
      N.CustomerCar_table();
      getch();
      break;
    case '4':
      clrscr();
      return;
    default:
      cout << '\a';
  }
  goto menu;
}
void status_editmenu() {
menu:
  clrscr();
  cout << "\n\t\t\tEDIT CAR STATUS\n\n";
  cout << "\n\t[1] EDIT All Status to [ AVAILABLE ] \n ";
  cout << "\n\t[2] EDIT All Status to [ BOOKED! ]\n";
  cout << "\n\t[3] Car Status Switch\n";
  cout << "\n\t[4] GO BACK\n";
  cout << "\nEnter your option => ";
  ventry(&::op);
  char carid[6];
  switch (::op) {
    case '1':
      clrscr();
      cout << '\a';
      status_edit("1");
      getch();
      break;
    case '2':
      clrscr();
      cout << '\a';
      status_edit("0");
      getch();
      break;
    case '3':
      clrscr();
      cout << '\a';
      cout << "\n\tEnter CarID to Update: ";
      gets(carid);
      status_edit(carid);
      getch();
      break;
    case '4':
      clrscr();
      return;
    default:
      cout << '\a';
  }
  goto menu;
}
void recordedit_menu() {
menu:
```

```cpp
  clrscr();
  border(3, 2, 74, 21, 2);
  border(23, 4, 37, 17, 0);
  gotoxy(27, 5);
  cout << "\tEDIT RECORD | MENU |";
  gotoxy(27, 8);
  cout << " [1] Update Car";
  gotoxy(27, 10);
  cout << " [2] Update Customer";
  gotoxy(27, 12);
  cout << " [3] DELETE Car";
  gotoxy(27, 14);
  cout << " [4] DELETE Customer";
  gotoxy(27, 16);
  cout << " [5] EDIT Status";
  gotoxy(27, 18);
  cout << " [6] GO BACK\n";
  gotoxy(1, 1);
  ventry(&::op);
  switch (::op) {
    case '1':
      clrscr();
      UpdateCar();
      break;
    case '2':
      clrscr();
      // UpdateCust();
      break;
    case '3':
      clrscr();
      DeleteCar();
      break;
    case '4':
      clrscr();
      DeleteCust();
      break;
    case '5':
      clrscr();
      status_editmenu();
      break;
    case '6':
      clrscr();
      return;
    default:
      cout << '\a';
  }
  goto menu;
}
void SearchDisplay_menu() {
menu:
  clrscr();
  border(3, 2, 74, 22, 2);
  border(23, 4, 37, 18, 0);
  gotoxy(27, 6);
  cout << "\t SEARCH AND DISPLAY";
  gotoxy(27, 10);
  cout << " [1] Display ALL Cars";
  gotoxy(27, 12);
  cout << " [2] Display ALL Customers";
  gotoxy(27, 14);
  cout << " [3] SEARCH Car";
  gotoxy(27, 16);
  cout << " [4] SEARCH Customer";
  gotoxy(27, 18);
  cout << " [5] GO BACK\n";
  gotoxy(1, 1);
  ventry(&::op);
  switch (::op) {
    case '1':
      clrscr();
      DisplayCar();
      break;
    case '2':
      clrscr();
      DisplayCust();
      break;
    case '3':
      clrscr();
      SearchCar_UI();
      break;
    case '4':
      clrscr();
      SearchCust_UI();
      break;
    case '5':
      clrscr();
      return;
    default:
      cout << '\a';
  }
  goto menu;
}
void menu() {
  clrscr();
  border(3, 2, 74, 22, 2);
// border(23,3,37,20,4);
menu:
  clrscr();
  // border(1,1,78,24,2);
  border(3, 2, 74, 22, 2);
  border(23, 3, 37, 20, 0);
  gotoxy(32, 4);
  cout << "/ M A I N   M E N U /";
  gotoxy(32, 5);
  cout << "--------------------";
  gotoxy(27, 7);
  cout << "\t[1] RENT A CAR";
  gotoxy(27, 9);
  cout << "\t[2] ADD CAR";
  gotoxy(27, 11);
  cout << "\t[3] SEARCH & DISPLAY";
  gotoxy(27, 13);
  cout << "\t[4] RETURN & BILLING";
  gotoxy(27, 15);
  cout << "\t[5] REPORTS";
  gotoxy(27, 17);
  cout << "\t[6] EDIT RECORD";
  gotoxy(27, 19);
  cout << "\t[7] ABOUT";
  gotoxy(27, 21);
  cout << "\t[8] EXIT\n";
  gotoxy(1, 1);
  ventry(&::op, 4, 1);
  switch (::op) {
    case '2':
      clrscr();
      AddCar();
      break;
    case '1':
      clrscr();
      AddCustomer();
      break;
    case '3':
      clrscr();
      SearchDisplay_menu();
      break;
    case '5':
      clrscr();
      report_menu();
      break;
    case '6':
      clrscr();
      recordedit_menu();
      break;
    case '4':
      clrscr();
      ReturnCar();
      break;
    case '7':
      clrscr();
      About();
      getch();
      break;
    case '8':
      clrscr();
      cout << "PROJECT HALTED";
      exit(-1);
    default:
      cout << '\a';
  }
  goto menu;
}
```

```
void main() {
  welcome();
  L.check();
  menu();
}
```