Mani Neeluri

Neeluru.mani@gmail.com

Neeluri Loka Manikanta

12210714

Roll No:72

K22BV

Laptop repair management

C Language

*Introduction:*

In today's world, laptops have become an essential part of our lives, and many individuals, including students and professionals, use them for their work. As a result, it's crucial to have a system that can manage the records of clients, laptops, and problems related to them. This report outlines the development of a system that can manage these records, including the functionalities of adding, updating, and deleting client and laptop records, describing problems, and assigning experts to resolve the issues.

Modules used in the program are:

1.Adding and storing client Information.

2.Adding and storing laptop Information.

3.Deleting\Updating Client Information.

4.Deleting\Updating Laptop Information.

5.Adding and storing Expert Information.

*Functionalities:*

The system allows the admin to perform various tasks such as:

1.Add a new client record by providing client information such as name, contact details, and any other relevant information.

2.Add a new laptop record by providing laptop information

such as make, model, serial number, and any other relevant information.

3.Update or delete an existing client record.

4.Update or delete an existing laptop record.

5.Add a description of the problem reported by the client and assign the problem to an expert who will resolve the issue.

6.View and manage the details of the experts who will resolve and assign the problem to an expert who will resolve the issue.

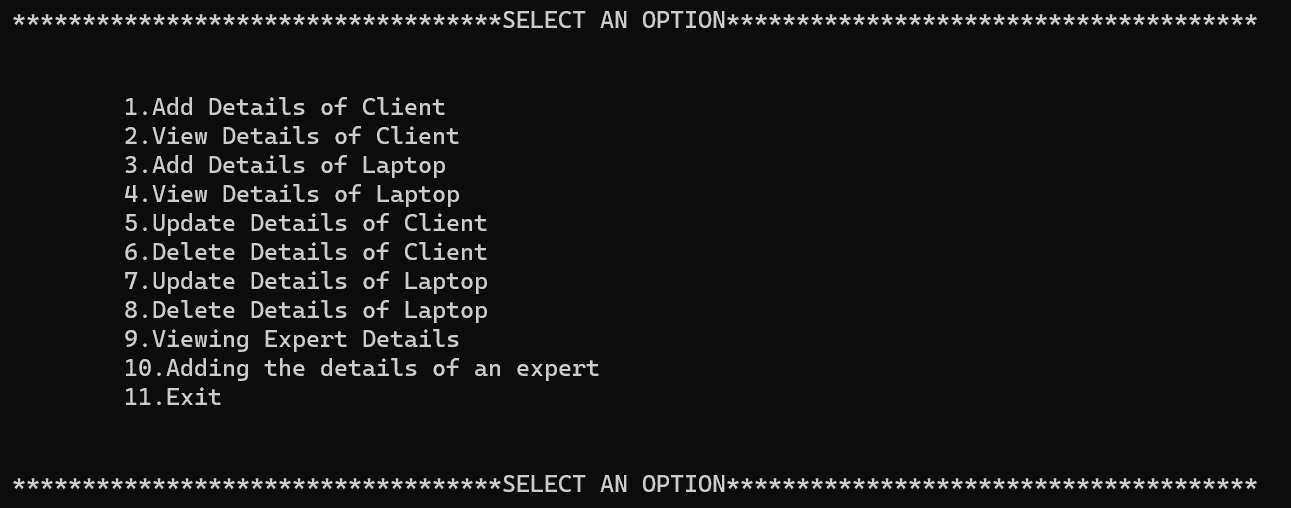
*Data Flow diagram:*

Diagram

Description automatically generated

*User Interface:*

Total of 11 options would be provided for the admin to edit, access and add the information and do-while concept is used in menu so that admin need not to run his program frequently and can exit the program after the completion of his work.



*Modules:*

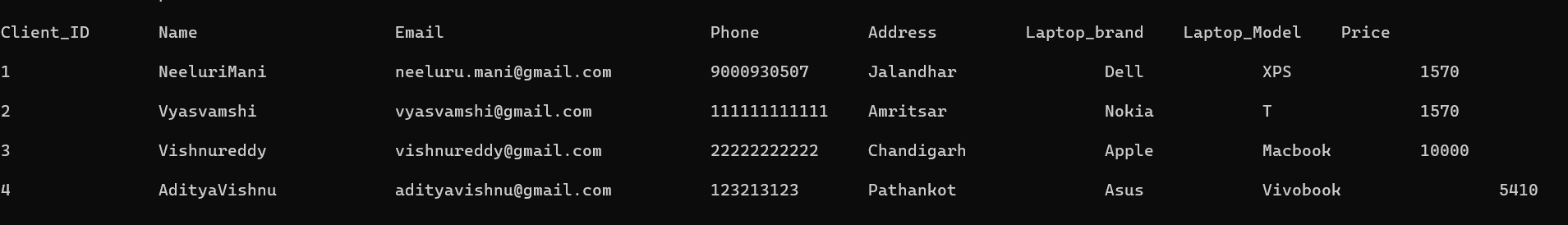
1.Client module:

Client module is capable enough of storing information of clients such as their id number, name , phone number, address, email id, basic information about their laptop and overall cost of their repair. Whole client data is stored in files so that it can be easily accessed and information would be safe concepts of file handling and functions such as fread, and fwrite are sued to store the information in a file. Admin can easily keep track of information of clients who had paid a visit for his shop for their laptop repair service.

Text

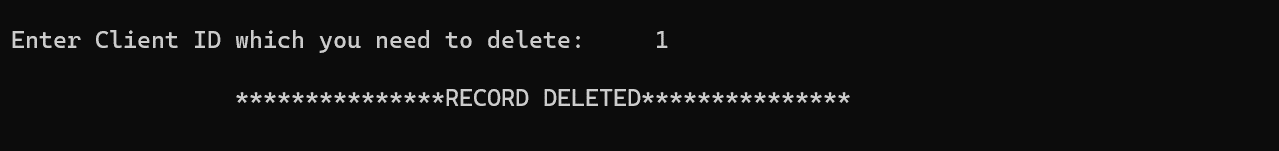
Description automatically generated

Viewing details of clients whose information is stored in text file with the help of file concepts.



Above Information can be updated and deleted so that admin can maintain accurate and consistent data in his system and can provide a better service for the clients with their accurate information.switch case is used in updating information so that only required information can be updated without typing whole information unnecessarily which would ultimately save a lot of time for the admin in saving time.Client ID would be sued here to update the information seamlessly and also wrong entrance of client id would be result in giving an error.A pop up list would be appeared so that admin can choose what data needs to be corrected in his database.switch case and temporary file concepts are used here so that file data can be easily changed.

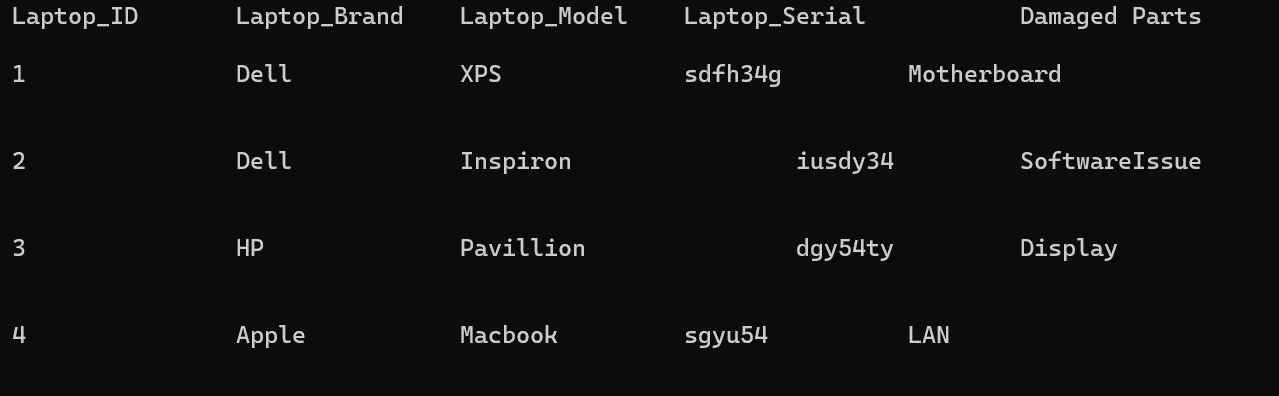
Text

Description automatically generatedClient Information could be easily modified and also delete facility is provided in this program for admins so that admin can delete the client information with the help of client id and and can maintain accurate data. All the data related to clients are stored in client.txt in their local computer.

*Laptop Module:*

Laptop module can store the information of laptop of users such as their laptop id, brand, model number, serial number, damaged parts if there are any and about the exact problem with the laptop that users are facing. Whole data about laptops are stored in laptop.txt file in their local computer folder where the program is saved by the admin.

Admin can view, add, and update the information of the data what they are having and can maintain consistent data and can provide a better service for the user. fwrite and fread functions are used to store the information in file and extract them to view them and update them.



Text

Description automatically generated

Deleting Information of a laptop from file using Laptop id.



*Expert Module:*

Expert module can be used to store the information of experts who can solve the problem of the laptop. Information such as expert id, expert name, skill, and their service charge is collected with the help of using arrays and they are stored in file using file concepts and functions such as fread and fwrite are used to read and write the information in files and can be helpful in viewing information.

Graphical user interface

Description automatically generated with medium confidence

Text

Description automatically generated

*Code:*

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct client {

int id;

char name[50];

char email[50];

char phone[15];

char address[250];

char laptop\_brand[50];

char laptop\_model[50];

int price;

}c;

struct laptop{

int id;

char brand[250];

char model[250];

char serial\_no[250];

char damaged\_parts[250];

}l;

struct expert{

int id;

char name[250];

int service\_cost;

char skill[250];

}e;

void add\_client(){

FILE \*a;

a=fopen("Client.txt","a");

printf("\n\n\n\tEnter ID of Client: ");

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

printf("\n\tEnter Name of client: ");

scanf("%s",&c.name);

printf("\n\tEnter Email-ID of client: ");

scanf("%s",c.email);

printf("\n\tEnter Phone Number of Client: ");

scanf("%s",&c.phone);

printf("\n\tEnter address of client: ");

scanf("%s",&c.address);

printf("\n\tEnter Laptop Brand: ");

scanf("%s",&c.laptop\_brand);

printf("\n\tEnter Model of Laptop: ");

scanf("%s",&c.laptop\_model);

printf("\n\tEnter the price to be paid by client: ");

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

printf("\n\t\t\t\t\t\*\*\*\*\*\*\*\*RECORD UPDATED\*\*\*\*\*\*\*\*\n\n\n");

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

fclose(a);

}

void view\_client(){

FILE \*b;

b=fopen("Client.txt","r");

printf("\nClient\_ID\tName\t\t\tEmail\t\t\t\tPhone\t\tAddress\t\tLaptop\_brand\tLaptop\_Model\tPrice\n\n");

while(fread(&c,sizeof(c),1,b)){

printf("%d\t\t%s\t\t%s\t\t%s\t%s\t\t%s\t\t%s\t\t%d\n\n",c.id,c.name,c.email,c.phone,c.address,c.laptop\_brand,c.laptop\_model,c.price);

}

fclose(b);

}

void add\_laptop(){

FILE \*c;

c=fopen("laptop.txt","a");

printf("\n");

printf("\n\n\tEnter the ID of Laptop: ");

scanf("%d",&l.id);

printf("\n\tEnter Laptop Brand: ");

scanf("%s",&l.brand);

printf("\n\tEnter Model of Laptop: ");

scanf("%s",&l.model);

printf("\n\tEnter serial number of laptop: ");

scanf("%s",&l.serial\_no);

printf("\n\tEnter damged parts of laptop: ");

scanf("%s",&l.damaged\_parts);

printf("\n\t\t\t\t\t\*\*\*\*\*\*\*\*RECORD UPDATED\*\*\*\*\*\*\*\*\n\n\n");

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

fclose(c);

}

void view\_laptop(){

FILE \*d;

d=fopen("laptop.txt","r");

printf("\nLaptop\_ID\tLaptop\_Brand\tLaptop\_Model\tLaptop\_Serial\t\tDamaged Parts\n\n");

while(fread(&l,sizeof(l),1,d)){

printf("%d\t\t%s\t\t%s\t\t%s\t\t%s\n\n\n",l.id,l.brand,l.model,l.serial\_no,l.damaged\_parts);

}

fclose(d);

}

void add\_expert(){

FILE \*d;

d=fopen("expert.txt","a");

printf("\n\n\n\tEnter the ID of Expert: ");

scanf("%d",&e.id);

printf("\n\tEnter the Name of Expert: ");

scanf("%s",&e.name);

printf("\n\tEnter the Skill of Expert: ");

scanf("%s",&e.skill);

printf("\n\tEnter service cost of expert: ");

scanf("%d",&e.service\_cost);

printf("\n\t\t\t\t\t\*\*\*\*\*\*\*\*RECORD UPDATED\*\*\*\*\*\*\*\*\n\n\n");

fwrite(&e,sizeof(e),1,d);

fclose(d);

}

void view\_expert(){

FILE \*e1;

e1=fopen("expert.txt","r");

printf("\nExpert\_ID\tName\t\tSkill\t\tService Cost\n\n\n");

while(fread(&e,sizeof(e),1,e1)){

printf("%d\t\t%s\t\t%s\t%d\n\n",e.id,e.name,e.skill,e.service\_cost);

}

}

int chec\_lid(int a){

FILE \*ni;

int c=0;

ni=fopen("laptop.txt","r");

while(!feof(ni)){

fread(&l,sizeof(l),1,ni);

if(a==l.id){

fclose(ni);

return 1;

}

}

fclose(ni);

return 0;

}

int chec\_client(int a){

FILE \*w1;

w1=fopen("client.txt","r");

while(!feof(w1)){

fread(&c,sizeof(c),1,w1);

if(a==c.id){

fclose(w1);

return 1;

}

}

fclose(w1);

return 0;

}

void update\_client(){

FILE \*d1;

FILE \*d2;

d1=fopen("client.txt","r");

d2=fopen("record.txt","w");

int avl;

int r,s,ch;

printf("\n\tEnter client ID which needs to be updated: ");

scanf("%d",&r);

avl=chec\_client(r);

if (avl==0){

printf("\n\t\t\t\*\*\*\*\*\*\*CLIENT ID NOT FOUND\*\*\*\*\*\*\*\n\n\n");

}

else{

while(fread(&c,sizeof(c),1,d1)){

s=c.id;

if(s!=r){

fwrite(&c,sizeof(c),1,d2);

}

else{

printf("\n\t\t\t1.Name");

printf("\n\t\t\t2.Email");

printf("\n\t\t\t3.Phone");

printf("\n\t\t\t4.Laptop\_brand");

printf("\n\t\t\t5.Laptop\_Model");

printf("\n\t\t\t6.Price");

printf("\n\t\t\t7.Address");

printf("\n\t\t\t8.Exit");

printf("\n\n\t\t\tEnter Your Choice: ");

scanf("%d",&ch);

switch(ch){

case(1):

printf("\n\t\t\tEnter Name: ");

scanf("%s",&c.name);

printf("\n\t\t\t\*\*\*\*\*\*\*\*\*\*RECORD UPDATED\*\*\*\*\*\*\*\*\*\*\n\n\n");

break;

case(2):

printf("\n\t\t\tEnter Email ID: ");

scanf("%s",&c.email);

printf("\n\t\t\t\*\*\*\*\*\*\*\*\*\*RECORD UPDATED\*\*\*\*\*\*\*\*\*\*\n\n\n");

break;

case(3):

printf("\n\t\t\tEnter Phone Number: ");

scanf("%s",&c.phone);

printf("\n\t\t\t\*\*\*\*\*\*\*\*\*\*RECORD UPDATED\*\*\*\*\*\*\*\*\*\*\n\n\n");

break;

case(7):

printf("\n\t\t\tEnter address of Client: ");

scanf("%s",&c.address);

break;

case(4):

printf("\n\t\t\tEnter Laptop Brand: ");

scanf("%s",&c.laptop\_brand);

printf("\n\t\t\t\*\*\*\*\*\*\*\*\*\*RECORD UPDATED\*\*\*\*\*\*\*\*\*\*\n\n\n");

break;

case(5):

printf("\n\t\t\tEnter Laptop Model: ");

scanf("%s",&c.laptop\_model);

printf("\n\t\t\t\*\*\*\*\*\*\*\*\*\*RECORD UPDATED\*\*\*\*\*\*\*\*\*\*\n\n\n");

break;

case(6):

printf("\n\t\t\tEnter new price needs to be paid by consumer: ");

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

printf("\n\t\t\t\*\*\*\*\*\*\*\*\*\*RECORD UPDATED\*\*\*\*\*\*\*\*\*\*\n\n\n");

break;

case(8):

exit(1);

break;

default:

printf("\n\t\t\t\*\*\*\*\*\*\*INVALID SELECTION\*\*\*\*\*\*\*");

break;

}

fwrite(&c,sizeof(c),1,d2);

}

}

fclose(d1);

fclose(d2);

d1=fopen("client.txt","w");

d2=fopen("Record.txt","r");

while(fread(&c,sizeof(c),1,d2)){

fwrite(&c,sizeof(c),1,d1);

}

fclose(d1);

fclose(d2);

}

}

void delete\_laptop(){

FILE \*fpo;

FILE \*fpt;

int r,s;

printf("\nEnter Laptop ID which you need to delete: ");

scanf("%d",&r);

if(chec\_lid(r)==0){

printf("\n\t\t\t\*\*\*\*\*\*\*\*\*\*LAPTOP ID NOT FOUND\*\*\*\*\*\*\*\*\*\*");

}

else{

fpo=fopen("Laptop.txt","r");

fpt=fopen("Record.txt","w");

while(fread(&l,sizeof(l),1,fpo)){

s=l.id;

if(s!=r){

fwrite(&l,sizeof(l),1,fpt);

}

}

fclose(fpo);

fclose(fpt);

fpo=fopen("Record.txt","r");

fpt=fopen("Laptop.txt","w");

while(fread(&l,sizeof(l),1,fpo)){

fwrite(&l,sizeof(l),1,fpt);

}

printf("\n\t\t\t\*\*\*\*\*\*\*\*\*\*RECORD DELETED\*\*\*\*\*\*\*\*\*\*\n\n\n\n");

fclose(fpo);

fclose(fpt);

}

}

void delete\_client(){

FILE \*fpo;

FILE \*fpt;

int r,s;

printf("\nEnter Client ID which you need to delete: ");

scanf("%d",&r);

if(chec\_client(r)==0){

printf("Client ID is not available in the file\n");

}

else{

fpo=fopen("Client.txt","r");

fpt=fopen("Record.txt","w");

while(fread(&c,sizeof(c),1,fpo)){

s=c.id;

if(s!=r){

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

}

}

fclose(fpo);

fclose(fpt);

fpo=fopen("Record.txt","r");

fpt=fopen("Client.txt","w");

while(fread(&c,sizeof(c),1,fpo)){

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

}

printf("\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RECORD DELETED\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\n");

fclose(fpo);

fclose(fpt);

}

}

void update\_laptop(){

int avl;

FILE \*fpt;

FILE \*fpo;

int s,r,ch;

printf("Enter Laptop ID to update: ");

scanf("%d",&r);

avl=chec\_lid(r);

if(avl==0){

printf("Laptop ID not found\n");

}

else{

fpo=fopen("laptop.txt","r");

fpt=fopen("tempfile.txt","w");

while(fread(&l,sizeof(l),1,fpo)){

s=l.id;

if(s!=r){

fwrite(&l,sizeof(l),1,fpt);

}

else{

printf("\n\t1.BRAND");

printf("\n\t2.MODEL");

printf("\n\t3.SERIAL NO");

printf("\n\t4.DAMAGED PARTS");

printf("\n\t6.EXIT");

printf("\nEnter your choice: ");

scanf("%d",&ch);

switch(ch){

case(1):

printf("Enter Brand: ");

scanf("%s",&l.brand);

break;

case(2):

printf("Enter Model: ");

scanf("%s",&l.model);

break;

case(3):

printf("Enter Serial No: ");

scanf("%s",&l.serial\_no);

break;

case(4):

printf("Enter Damaged Parts: ");

scanf("%s",l.damaged\_parts);

break;

break;

default:

printf("Invalid Selection");

break;

}

fwrite(&l,sizeof(l),1,fpt);

}

}

fclose(fpo);

fclose(fpt);

fpo=fopen("laptop.txt","w");

fpt=fopen("tempfile.txt","r");

while(fread(&l,sizeof(l),1,fpt)){

fwrite(&l,sizeof(l),1,fpo);

}

fclose(fpo);

fclose(fpt);

printf("RECORD UPDATED\n");

}

}

int main(){

int option;

do{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SELECT AN OPTION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\n");

printf("\t1.Add Details of Client\n");

printf("\t2.View Details of Client\n");

printf("\t3.Add Details of Laptop\n");

printf("\t4.View Details of Laptop\n");

printf("\t5.Update Details of Client\n");

printf("\t6.Delete Details of Client\n");

printf("\t7.Update Details of Laptop\n");

printf("\t8.Delete Details of Laptop\n");

printf("\t9.Viewing Expert Details\n");

printf("\t10.Adding the details of an expert\n");

printf("\t11.Exit\n\n\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SELECT AN OPTION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\nPlease enter a option: ");

scanf("%d",&option);

switch(option){

case(1):

add\_client();

break;

case(2):

view\_client();

break;

case(3):

add\_laptop();

break;

case(4):

view\_laptop();

break;

case(5):

update\_client();

break;

case(6):

delete\_client();

break;

case(7):

update\_laptop();

break;

case(8):

delete\_laptop();

break;

case(9):

view\_expert();

break;

case(10):

add\_expert();

break;

case(11):

exit(1);

break;

default:

printf("Choose valid option");

}}

while(option!=11);

}

*Conclusion:*

The system we have developed allows the admin to manage the records of clients, laptops, and problems efficiently. The system's various modules enable the admin to add, update, and delete client and laptop records, describe problems, assign experts to resolve the issues, and generate reports based on various criteria. This system can be beneficial for any organization that deals with laptop repairs or maintenance, making the process smoother and more manageable.