DSA [20ES117] COURSE PROJECT REPORT

On

“Internet Cafe Management System”

Developed By:

H.T.NO	STUDENT NAME
2103A51173	Mallikarjun Marthi

Under the Guidance of

Dr. A. SIVA KRISHNA REDDY

Associate Professor

Submitted to

Department Computer Science and Artificial Intelligence

SR University

Ananthasagar(V), Hasanparthy(M), Hanamkonda (Dist.) – 506371

www.sru.edu.in

December 2022

Department of Computer Science and Artificial Intelligence

CERTIFICATE

This is to certify that the DSA course project report entitled **“Internet Cafe Management System”** is a record of bonafide work carried out by the student **Mallikarjun Marthi** bearing roll number **2103A51173** of Computer Science and Artificial Intelligence department during the academic year 2022-23.

Supervisor

(Dr. A. Siva Krishna Reddy)

INDEX

IDEA

- My idea is to provide an application that uses the data structures that provide the platform to the internet cafe to store the information of the users in a file.
- It also helps the admin to track the users information daily and for each session.

PROBLEM STATEMENT:

Develop a C application to dynamically read and store details of ‘n’ customers and admin information in terms of their user- name ,user id, password, their feedback, the registration time and account balance. The application also provides the users with the following mentioned functionalities:

1.Menu selection for Admin or User selection.

2.Login credentials for users to access the computers in cafe.

3.Read data of ‘n’ users dynamically (registration) .

4.Take the feedback from the user for admin purposes.

5.Addition of users into the files

6.Viewing the history of previous session login

7.Deletion of specified user whenever needed.

8.File updating to record the present session activities.

And many more good functions for highly efficient system performance whenever any one of the above-mentioned menu functions are executed the respective menu options of either the user or admin are displayed iteratively.

MODULES:

In this application all variables and structure are declared globally so that these variables and structure members can be accessed throughout the program at any function call. We can choose any function by using function calls which are declared in switch-case. In order to repeat the loop control statement (while) is used with condition. The memory allocation will be done in this program dynamically.

In this application following modules are used.

1. Registration

In this module, the application takes the data of the customer of our internet cafe .The application asks the fields like username and pass word the user wants to keep for further login and feedback purposes.The system prints the user id and the available account balance in the cafe.

2. Login

In this module, The user has to enter the userid provided when he registered in the cafe followed by passoword he has chosen for his account.The login is successful only when both the fields are correctly entered and if userid is wrongly entered he has to register and if login is success he is redirected to billing page at the time of logout.The billing action after login is a mandatory task.

3. Feedback

In this module, the application asks the customer to give his valuable message to give any grievances or concerns to regarding the services.

4.Change credentials

In this module, the application allows the admin to change the credentials. The admin has to provide his previous credentials to confirm he is the owner and then he can change the credentials.

5.Delete user

In this module, the application allows the admin to delete the specific user taking the user id as the input. It then deletes the user entirely from the record.

6.Review

In this module, the application prints the following details of the users in the particular session : UserName ,UserId,Balance,Feedback & registration time of users.

7.History

In this application the application shows or prints the data of the all the previous sessions of the system starting from the day its used. It prints the session wise users data as like review .It helps the admin to track users.

8.File updating

In this module ,the application updates the cafe record with the present session's users' information.The users fields like the Name, User Id, Registration Time, Feedback and balance are added to the existing record of the cafe.

Usage of Tools

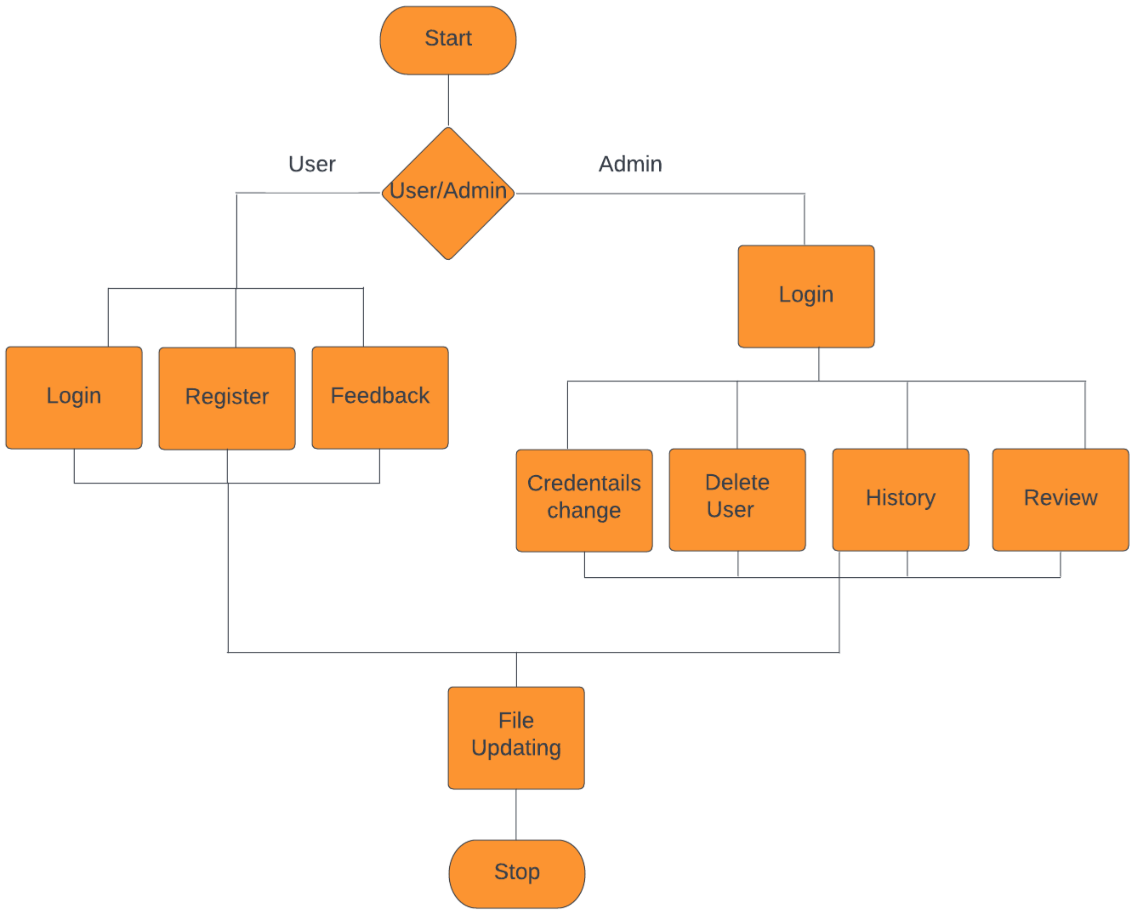
Hardware Tools:

Operating System	Windows 2000/NT/XP/Vista
Ram	256 MB or more
HARD DISK	40 GB or more
Processor	P3 or High

Software Tools:

Dev-C++ is a free full-featured integrated development environment distributed under the GNU General Public License for programming in C and C++. It is written in Delphi. It is bundled with, and uses, the MinGW or TDM-GCC 64bit port of the GCC as its compiler.

Block Diagram:



KNOWLEDGE REQUIRED TO DEVELOP THIS APPLICATION

- Control Statements (if, if-else, switch)
- Loop Statements
- Arrays (1-arrays)
- Strings (Strings) and its functions (strcpy , strcmp)
- Functions (Any type of user defined functions)
- Structures (structures and nested structures)
- Pointers (pointer to strings and pointers to structures)
- Dynamic Memory Allocation (malloc/ calloc/ realloc)
- Linked Lists (Single Linked List)
- Time and windows libraries functions

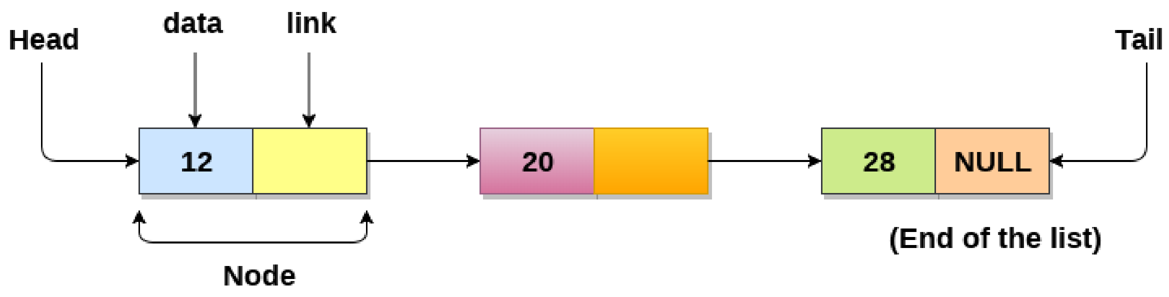
Solution

- ✍ The application can be implemented in internet cafes in order to manage its customers and their information.
- ✍ Firstly the interface takes the input such as their Names, Desired password dynamically from the users and registers them as the customers of the cafe.
- ✍ The application also provides a structured manner information of its users in the form of their Name, User Id, Feedback, Registration time and their account balance in the cafe.
- ✍ This interface gives the customers the equal authority as the owner(Admin) by allowing them give the feedback or grievances.
- ✍ Here the data structures Linked lists can provide us the functionalities we require to register or login or delete an user.

Data Structures:

Single Linked List:

A linked list is a linear data structure that includes a series of connected nodes. Here, each node stores the data and the address of the next node. For example,



Linked List can be defined as collection of objects called nodes that are randomly stored in the memory.

A node contains two fields i.e. data stored at that particular address and the pointer which contains the address of the next node in the memory.

The last node of the list contains pointer to the null.

Representation of Linked List:

Let's see how each node of the linked list is represented. Each node consists:

- A data item
- An address of another node

We wrap both the data item and the next node reference in a struct as:

```
struct node
{
int data;
struct node *next;
}
struct node *head=NULL;
```

Linked List Applications:

- Dynamic memory allocation
- Implemented in stack and queue
- In **undo** functionality of softwares
- Hash tables, Graphs

Uses of Linked Lists:

- The list is not required to be contiguously present in the memory. The node can reside anywhere in the memory and linked together to make a list. This achieves optimized utilization of space.
- List size is limited to the memory size and doesn't need to be declared in advance.
- Empty node cannot be present in the linked list.
- We can store values of primitive types or objects in the singly linked list.

Why use linked list over array?

Till now, we were using array data structure to organize the group of elements that are to be stored individually in the memory. However, Array has several advantages and disadvantages which must be known in order to decide the data structure which will be used throughout the program.

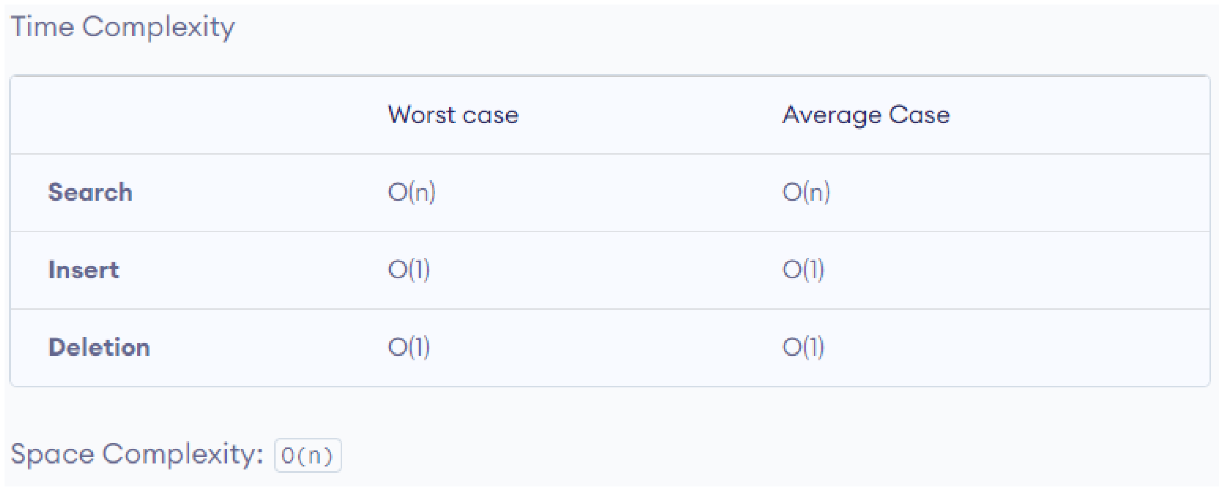
Array contains following limitations:

1. The size of array must be known in advance before using it in the program.
2. Increasing size of the array is a time taking process. It is almost impossible to expand the size of the array at run time.
3. All the elements in the array need to be contiguously stored in the memory. Inserting any element in the array needs shifting of all its predecessors.

Linked list is the data structure which can overcome all the limitations of an array. Using linked list is useful because,

1. It allocates the memory dynamically. All the nodes of linked list are non-contiguously stored in the memory and linked together with the help of pointers.
2. Sizing is no longer a problem since we do not need to define its size at the time of declaration. List grows as per the program's demand and limited to the available memory space.

Linked List Complexity:



Time Complexity

	Worst case	Average Case
Search	O(n)	O(n)
Insert	O(1)	O(1)
Deletion	O(1)	O(1)

Space Complexity: `O(n)`

Linked List Operations:

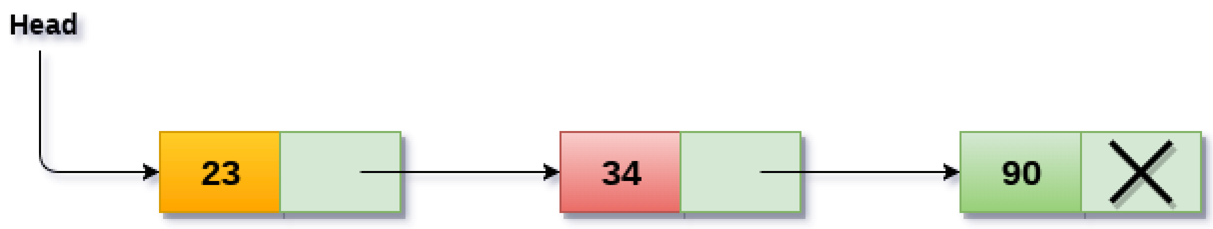
Here's a list of basic linked list operations that we will cover in this article.

- Traversal - access each element of the linked list
- Insertion - adds a new element to the linked list
    - Insertion can be done at beginning, ending and after a specific node
- Deletion - removes the existing elements
    - Deletion can be done at the beginning , ending and after a specific node.

Traverse a Linked List:

Traversing is the most common operation that is performed in almost every scenario of singly linked list. Traversing means visiting each node of the list once in order to perform some operation on that. This will be done by using the following statements.

```
ptr = head;
        while (ptr!=NULL)
            {
                ptr = ptr -> next;
            }
```



Algorithm:

```
o STEP 1: SET PTR = HEAD
o STEP 2: IF PTR = NULL
    WRITE "EMPTY LIST"
    GOTO STEP 7
    END OF IF
o STEP 4: REPEAT STEP 5 AND 6 UNTIL PTR != NULL
o STEP 5: PRINT PTR→ DATA
o STEP 6: PTR = PTR → NEXT
    [END OF LOOP]
o STEP 7: EXIT
```

Insertion of Element in Single Linked List at Beginning:

Inserting a new element into a singly linked list at beginning is quite simple. We just need to make a few adjustments in the node links. There are the following steps which need to be followed in order to inser a new node in the list at beginning.

- Allocate the space for the new node and store data into the data part of the node. This will be done by the following statements.

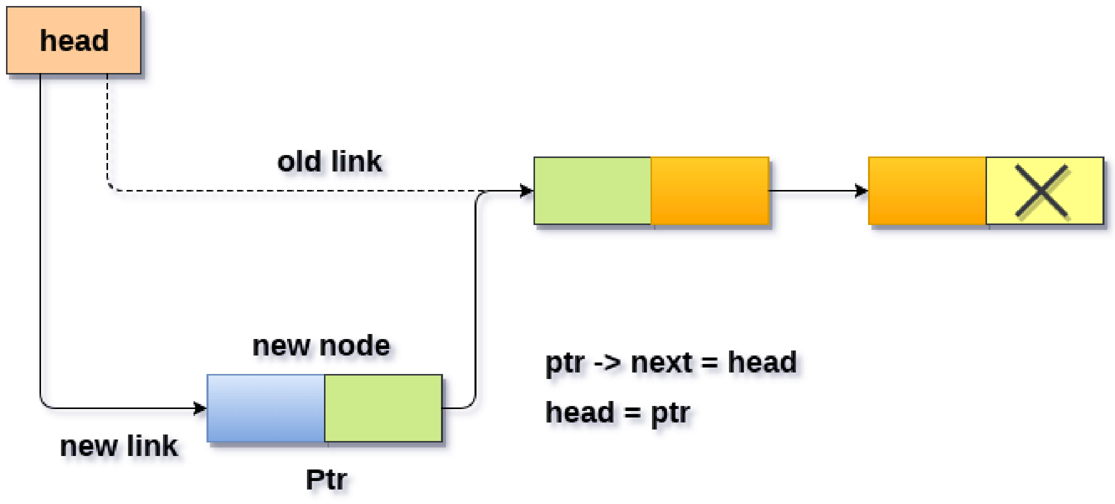
1. ptr = (struct node *) malloc(sizeof(struct node *));
2. ptr → data = item

- Make the link part of the new node pointing to the existing first node of the list. This will be done by using the following statement.

1. ptr->next = head;

- At the last, we need to make the new node as the first node of the list this will be done by using the following statement.

1. head = ptr;



Algorithm:

```
○ Step 1: IF PTR = NULL
          Write OVERFLOW
          Go to Step 7
          [END OF IF]
○ Step 2: SET NEW_NODE = PTR
○ Step 3: SET PTR = PTR → NEXT
○ Step 4: SET NEW_NODE → DATA = VAL
○ Step 5: SET NEW_NODE → NEXT = HEAD
○ Step 6: SET HEAD = NEW_NODE
○ Step 7: EXIT
```

Insertion of Element in Single Linked List at the End:

In order to insert a node at the last, there are two following scenarios which need to be mentioned.

1. The node is being added to an empty list
2. The node is being added to the end of the linked list

in the first case,

- The condition (head == NULL) gets satisfied. Hence, we just need to allocate the space for the node by using malloc statement in C. Data and the link part of the node are set up by using the following statements.

```
1. ptr->data = item;
2. ptr -> next = NULL;
```

- Since, **ptr** is the only node that will be inserted in the list hence, we need to make this node pointed by the head pointer of the list. This will be done by using the following Statements.

```
1. Head = ptr
```

In the second case,

- The condition **Head = NULL** would fail, since Head is not null. Now, we need to declare a temporary pointer temp in order to traverse through the list. **temp** is made to point the first node of the list.

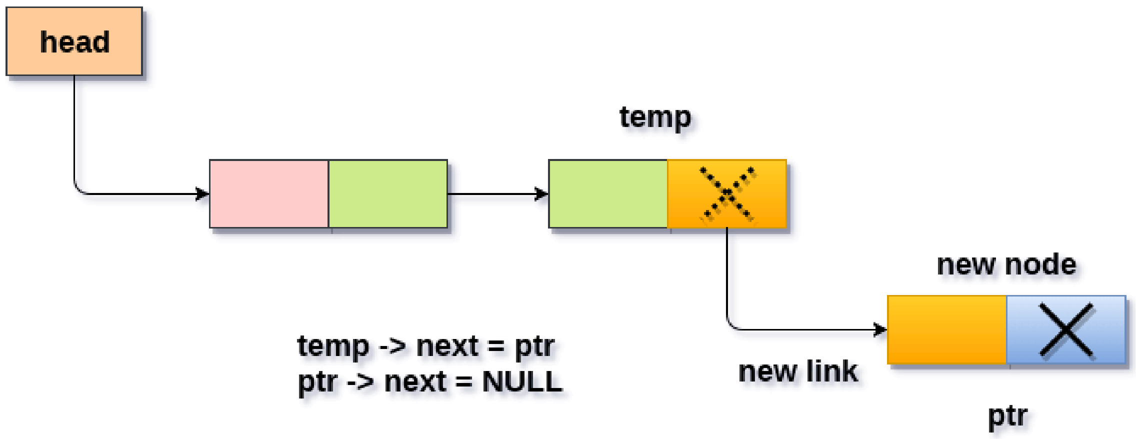
```
1. Temp = head
```

- Then, traverse through the entire linked list using the statements:

```
1. while (temp→ next != NULL)
2.             temp = temp → next;
```

- At the end of the loop, the temp will be pointing to the last node of the list. Now, allocate the space for the new node, and assign the item to its data part. Since, the new node is going to be the last node of the list hence, the next part of this node needs to be pointing to the **null**. We need to make the next part of the temp node (which is currently the last node of the list) point to the new node (ptr) .

```
1. temp = head;
2.         while (temp -> next != NULL)
3.         {
4.             temp = temp -> next;
5.         }
6.         temp->next = ptr;
7.         ptr->next = NULL;
```



Algorithm:

```
o Step 1: IF PTR = NULL Write OVERFLOW
      Go to Step 1
     [END OF IF]

o Step 2: SET NEW_NODE = PTR

o Step 3: SET PTR = PTR - > NEXT

o Step 4: SET NEW_NODE - > DATA = VAL

o Step 5: SET NEW_NODE - > NEXT = NULL

o Step 6: SET PTR = HEAD

o Step 7: Repeat Step 8 while PTR - > NEXT != NULL

o Step 8: SET PTR = PTR - > NEXT
  [END OF LOOP]

o Step 9: SET PTR - > NEXT = NEW_NODE

o Step 10: EXIT
```

Insertion of Element in Single Linked List after Specified Node:

- In order to insert an element after the specified number of nodes into the linked list, we need to skip the desired number of elements in the list to move the pointer at the position after which the node will be inserted. This will be done by using the following statements.

```
temp=head;
        for(i=0;i<loc;i++)
        {
           temp = temp->next;
           if(temp == NULL)
           {
              return;
           }

               }
```

- Allocate the space for the new node and add the item to the data part of it. This will be done by using the following statements.

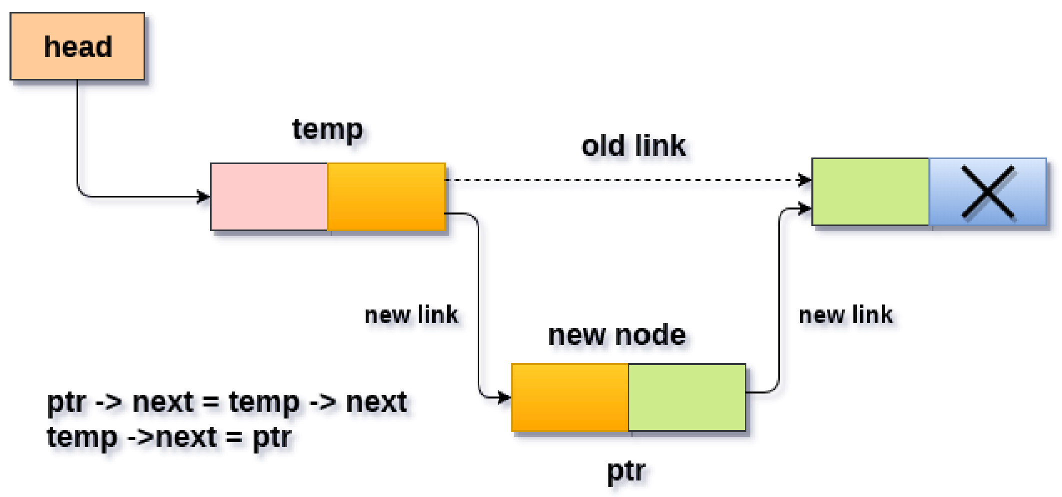
```
ptr = (struct node *) malloc (sizeof(struct node));
      ptr->data = item;
```

- Now, we just need to make a few more link adjustments and our node at will be inserted at the specified position. Since, at the end of the loop, the loop pointer temp would be pointing to the node after which the new node will be inserted. Therefore, the next part of the new node ptr must contain the address of the next part of the temp (since, ptr will be in between temp and the next of the temp). This will be done by using the following statements.

```
ptr→ next = temp → next
```

- Now, we just need to make the next part of the temp, point to the new node ptr. This will insert the new node ptr, at the specified position.

```
temp ->next = ptr;
```



Algorithm:

- **STEP 1:** IF PTR = NULL

```
WRITE OVERFLOW
    GOTO STEP 12
   END OF IF
```

- **STEP 2:** SET NEW_NODE = PTR
- **STEP 3:** NEW_NODE → DATA = VAL
- **STEP 4:** SET TEMP = HEAD
- **STEP 5:** SET I = 0
- **STEP 6:** REPEAT STEP 5 AND 6 UNTIL I<loc< li=""></loc<>
- **STEP 7:** TEMP = TEMP → NEXT
- **STEP 8:** IF TEMP = NULL

```
WRITE "DESIRED NODE NOT PRESENT"
     GOTO STEP 12
    END OF IF
 END OF LOOP
```

- **STEP 9:** PTR → NEXT = TEMP → NEXT
- **STEP 10:** TEMP → NEXT = PTR
- **STEP 11:** SET PTR = NEW_NODE
- **STEP 12:** EXIT

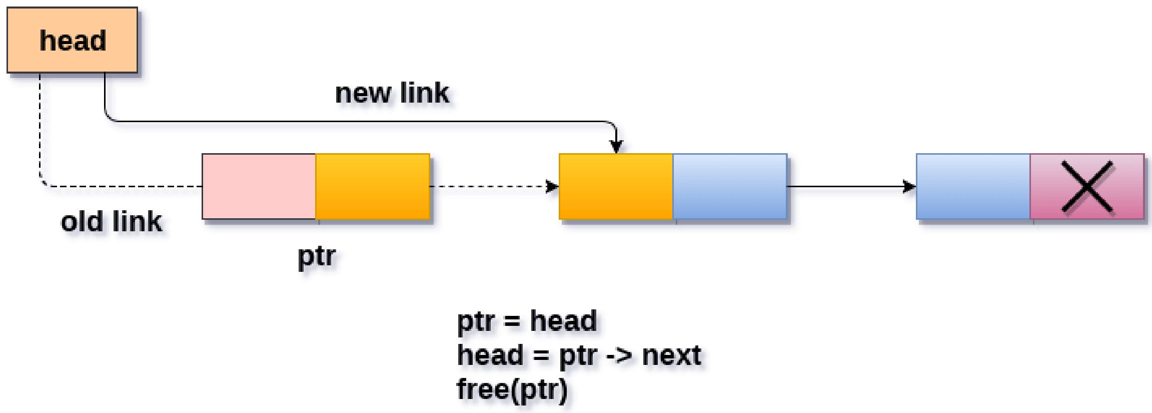
Deletion of Element in Single Linked List at Beginning:

Deleting a node from the beginning of the list is the simplest operation of all. It just need a few adjustments in the node pointers. Since the first node of the list is to be deleted, therefore, we just need to make the head, point to the next of the head. This will be done by using the following statements.

1. ptr = head;
2. head = ptr->next;

Now, free the pointer ptr which was pointing to the head node of the list. This will be done by using the following statement.

1. free(ptr)



Algorithm:

- **Step 1:** IF HEAD = NULL

```
Write UNDERFLOW
     Go to Step 5
    [END OF IF]
```

- **Step 2:** SET PTR = HEAD
- **Step 3:** SET HEAD = HEAD -> NEXT
- **Step 4:** FREE PTR
- **Step 5:** EXIT

Deletion of Element in Single Linked List at the End:

There are two scenarios in which, a node is deleted from the end of the linked list.

1. There is only one node in the list and that needs to be deleted.
2. There are more than one node in the list and the last node of the list will be deleted.

In the first scenario,

the condition head → next = NULL will survive and therefore, the only node head of the list will be assigned to null. This will be done by using the following statements.

1. ptr = head
2. head = NULL
3. free(ptr)

In the second scenario,

The condition head → next = NULL would fail and therefore, we have to traverse the node in order to reach the last node of the list.

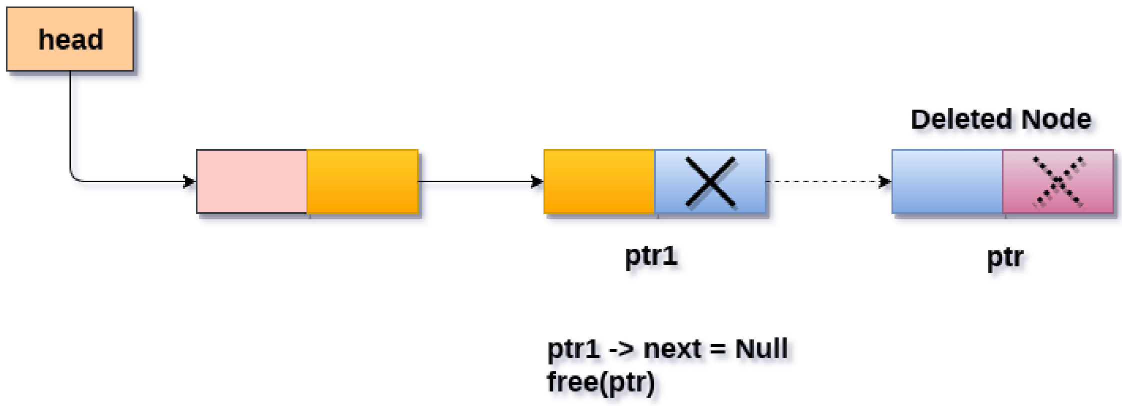
For this purpose, just declare a temporary pointer temp and assign it to head of the list. We also need to keep track of the second last node of the list. For this purpose, two pointers ptr and ptr1 will be used where ptr will point to the last node and ptr1 will point to the second last node of the list.

this all will be done by using the following statements.

```
1. ptr = head;
2.            while(ptr->next != NULL)
3.            {
4.                ptr1 = ptr;
5.                ptr = ptr ->next;
6.            }
```

Now, we just need to make the pointer ptr1 point to the NULL and the last node of the list that is pointed by ptr will become free. It will be done by using the following statements.

```
1. ptr1->next = NULL;
2. free(ptr);
```



Algorithm:

```
○ Step 1: IF HEAD = NULL

Write UNDERFLOW
   Go to Step 8
  [END OF IF]

○ Step 2: SET PTR = HEAD

○ Step 3: Repeat Steps 4 and 5 while PTR -> NEXT!= NULL

○ Step 4: SET PREPTR = PTR

○ Step 5: SET PTR = PTR -> NEXT

[END OF LOOP]

○ Step 6: SET PREPTR -> NEXT = NULL

○ Step 7: FREE PTR

○ Step 8: EXIT
```

Deletion of Element in Single Linked List after the Specified Node:

In order to delete the node, which is present after the specified node, we need to skip the desired number of nodes to reach the node after which the node will be deleted. We need to keep track of the two nodes. The one which is to be deleted the other one if the node which is present before that node. For this purpose, two pointers are used: ptr and ptr1.

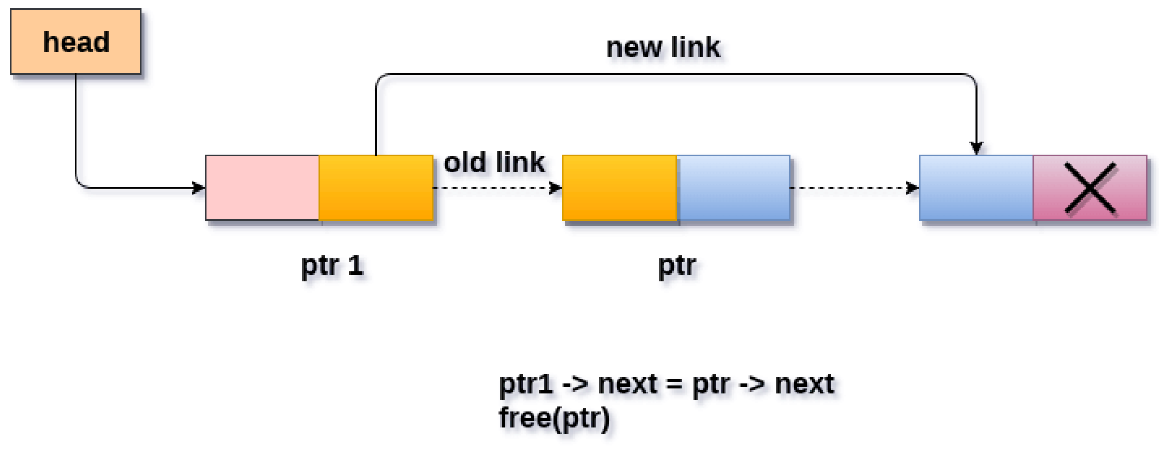
Use the following statements to do so.

```
ptr=head;
        for(i=0;i<loc;i++)
        {
            ptr1 = ptr;
            ptr = ptr->next;

            if(ptr == NULL)
            {
                printf("\nThere are less than %d elements in the list..",loc);
                        return;
                    }
                }
```

Now, our task is almost done, we just need to make a few pointer adjustments. Make the next of ptr1 (points to the specified node) point to the next of ptr (the node which is to be deleted).

This will be done by using the following statements.

```
ptr1 ->next = ptr ->next;
        free(ptr);
```



Algorithm:

```
o STEP 1: IF HEAD = NULL

WRITE UNDERFLOW
    GOTO STEP 10
   END OF IF

o STEP 2: SET TEMP = HEAD

o STEP 3: SET I = 0

o STEP 4: REPEAT STEP 5 TO 8 UNTIL I<loc< li=""></loc<>

o STEP 5: TEMP1 = TEMP

o STEP 6: TEMP = TEMP → NEXT

o STEP 7: IF TEMP = NULL

WRITE "DESIRED NODE NOT PRESENT"
    GOTO STEP 12
    END OF IF

o STEP 8: I = I+1

END OF LOOP

o STEP 9: TEMP1 → NEXT = TEMP → NEXT

o STEP 10: FREE TEMP

o STEP 11: EXIT
```

SOURCE CODE:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include <windows.h>

#define MAX_SIZE 50

/*
 Developer : M.Mallikarjun
 Date      :31-10-2022
 Application : Internet Cafe Management system.
 Libraries Used:
 1.STDIO
 2.STDLIB
 3.MATHS
 4.TIME
 5.WINDOWS
 Static Declared variables a----->User ids
 global variable structures for user and admin.
 */

void user(); // user defined function for customers.
int admin(); //Admin side functional option.

// ADMIN OPTIONS.
void change();
void add();
void del();
int bill();
```

```
void review();
void DelUser();
void history();

//USER OPTIONS.
int log_in();
void reg();
void cancel();
void feedback();

//file updation
void file_update();

// HEADINGS OF EACH PORTION.
void heading()
{
time_t now=time(NULL);
char *st=ctime(&now);
   printf("\n\t\t\t\t-------------------------------------------------\t\t\t\t");
   printf("\n\t\t\t\t|   WELCOME TO INTERNET CAFE MANAGEMENT SYSTEM   |\t\t");
   printf("\n\t\t\t\t|    - The place where world under your fingers  |\t\t");

          printf("\n\t\t\t\t-------------------------------------------------\t\t\t\t");
   printf("\n\t\t\t\t      %s                              \t\t",st);
   //printf("\t\t\t\t-------------------------------------------------\t\t\t\t\n");
}

void main_heading()
{
time_t now=time(NULL);
char *st=ctime(&now);
   printf("\n\t\t\t\t-------------------------------------------------\t\t\t\t");
   printf("\n\t\t\t\t|        INTERNET CAFE MANAGEMENT SYSTEM       |\t\t");
```

```
    printf("\n\t\t\t\t|        --Every click you make brings you     |\t\t");
    printf("\n\t\t\t\t|          close to our hearts    %c          |\t\t",1);

    printf("\n\t\t\t\t----------------------------------------------------------\t\t\t\t");
          printf("\n\t\t\t\t----------------------------------------------------------------\t\t\t\t");
    printf("\n\t\t\t\t     %s                              \t\t",st);
    //printf("\n\t----------------------------------------------------------------\t\t\t\t\n");
}

//Time module to calculate the users registration time.
//declaring the linked list to store the users data;


//Structures to store the users.
struct user
{
char name[150];
int user_id;
char feedback[4000];
char userpass[MAX_SIZE];
struct user *next;
char reg_time[MAX_SIZE];
int balance;
};
struct user *first=NULL;


// static defining the admin name and password.
char owner[MAX_SIZE]="OWNER";
char pass[MAX_SIZE]="PASS";


// ADMIN INFORMATION STORING STRUCTURE.
```

```
struct admin
 {
        char user_name[MAX_SIZE];
        char password[MAX_SIZE];
 };
struct admin *ad;



int main()
{



int ch;
while(1)
{
  system("color CF");
        main_heading();
        printf("\n\t\t\t\t\t\t 1.USER \n\t\t\t\t\t\t 2.ADMIN \n\t\t\t\t\t\t 3.exit");
        printf("\n\t\t\t\t\t\t\t enter the choice :\t");
        scanf("%d",&ch);

        system("cls");
        switch(ch)
        {
                case 1:
                                {
                                        user();
                                        break;
                                }
                case 2:
                        {
                                admin();
                                break;
                        }
                case 3:
                        {
                                file_update();
                                exit(0);
                                break;
                        }
                default:
                        printf("\n\t\t\t\t\t\tPLEASE CHECK YOUR INPUT AND ENTER A
VALID INPUT");
        }

}

return 0;
}
void user()
{
heading();
        int ch;
        while(1)
        {       printf("\n\t\t\t\t\t\t YOUR AVAILABLE OPTIONS ARE:");
                printf("\n\t\t\t\t\t\t1.Login");
                printf("\n\t\t\t\t\t\t2. NEW REGISTRATION");
                printf("\n\t\t\t\t\t\t3. FEEDBACK");
                printf("\n\t\t\t\t\t\t4. EXIT");
                printf("\n");
                printf("\n\t\t\t\t\t\t enter the choice");
                scanf("%d",&ch);
                system("cls");

                switch(ch)
                {
                        case 1:
                                heading();
                                log_in();
                                break;
                        case 2:
                                heading();
                                reg();
                                break;
                        case 3:
                                heading();
                                feedback();
                                break;
                        case 4:
                                return ;
                        default:
                                printf("\n PLEASE ENTER A VALID INPUT");
                }
        }
        system("cls");
}


int admin()
{       int namecmp,passcmp;
struct admin *ad=(struct admin*)malloc(sizeof(struct admin));
heading();
printf("\n\t\t\t\t\t\t LOGIN TO CONTINUE\n\t");
while(1)
{

        printf("\n\t\t\t\t\t\t PLEASE ENTER THE USER NAME \t");
        fflush(stdin);
        scanf("%s",(ad->user_name));
        printf("\n\t\t\t\t\t\t PLEASE ENTER THE PASSWORD\t");
        fflush(stdin);
        scanf("%s",ad->password);
        namecmp=strcmp(ad->user_name,owner);
        passcmp=strcmp(ad->password ,pass);
        if(namecmp !=0 || passcmp!=0)
        {

                printf("\n\t\t\t\t\t\t PLEASE ENTER THE CORRECT CREDENTIALS");
                continue;
        }
        else
        {
                printf("\n\t\t\t\t\t\t YOU HAVE SUCCESSFULLY LOGGED IN ");
                int ch;
                while(1)
                {
                        printf("\n\t\t\t\t\t\t DEAR ADMIN CHOOSE YOUR OPTION TO
PERFORM");
                        printf("\n\t\t\t\t\t\t 1.CHANGE YOUR CREDENTIALS");
                        printf("\n\t\t\t\t\t\t 2.DELETE THE USER");
                        printf("\n\t\t\t\t\t\t 3.REVIEW BOX");
                        printf("\n\t\t\t\t\t\t 4.HISTORY");
                        printf("\n\t\t\t\t\t\t5.EXIT");
                        scanf("%d",&ch);
                        switch(ch)
                                {
                                        case 1:
                                                {
                                                        heading();
                                                        change();
                                                        break;
                                                }
                                        case 2:
                                                {
                                                        heading();
                                                        del();
                                                        break;
                                                }

                                        case 3:
                                                heading();
                                                review();
                                                break;
                                        case 0:
                                                add();

                                        case 4:
                                                {
                                                        history();
                                                        break;
                                                }
                                        case 5:
                                                {
                                                        main();
                                                }
                                        default:
                                                printf("\n\t\t\t\t\t\t PLEASE CHOOSE A
VALID INPUT");
                                }
                        }
                }
}
                system("cls");
```

```
}

void reg()
{

printf("\n\t\t\t\t\t PLEASE CONTACT ADMIN AND PROVIDE THE REQ DETAILS\t");




printf("\n\t\t\t\t\t\t------------\n");
printf("\n\t\t\t\t\t\t|%s|\n",__DATE__);
printf("\n\t\t\t\t\t\t|%s|",__TIME__);
printf("\n\t\t\t\t\t\t------------\n");
add();

}

void add()
{       system("cls");
struct user *temp,*newuser=(struct user*)malloc(sizeof(struct user));
char name[MAX_SIZE],pass[MAX_SIZE];
static  int a=1000;
register int x,flag;
main_heading();
printf("\n\t\t\t\t THE RECORDS OF PREVIOUS USERS ARE:\n");
printf("\n\t\t\t\t\tUsername:\t\t UserId");

        for(temp=first;temp!=NULL;temp=temp->next)
        {
        printf("\n\t\t\t\t-------------------------------------------------------------\n");
        printf("\n\t\t\t\t\t%s\t\t%d",temp->name,temp->user_id);
        }
```

```
	printf("\n\t\t\t\t-------------------------------------------------------------\n");

printf("\n\t\t\t\t\t\t Dear, provide the user details");
printf("\n\t\t\t\t\t\t ENTER THE FULL NAME\t");
scanf("%s",newuser->name);
fflush(stdin);
printf("\n\t\t\t\t\t\t ENTER THE PASSWORD\t");
scanf("%s",newuser->userpass);
newuser->next=NULL;
for(temp=first;temp!=NULL;temp=temp->next)
{
	x=strcmp(temp->name,newuser->name);
	if(x==0)
		flag=0;
	else
		flag=1;
}
if(flag!=0)
{

	if(first == NULL)
	  first = newuser;
	else
	{	temp=first;
		while(temp->next!=NULL)
		{
			temp=temp->next;
		}
		temp->next=newuser;

	}
	time_t login=time(NULL);
	char *reg_time=ctime(&login);
```

```
        newuser->user_id= ++a;
        strcpy(newuser->reg_time,reg_time);
        printf("\n\t\t\t\t\t\tYOUR client has been added successfully\t");
        printf("\n\t\t\t\t\t\tDear User!! your id is :%d",newuser->user_id);
        newuser->balance=1000;
        printf("\n\t\t\t\t\t\tDear user ! you have an initial balance of 1000 amount");

}
else
{
        printf("\n\t\t\t\t\t Username already existing");
        add();
}
sleep(2);
system("cls");
}

int bill(int login,int uid) //Done.
{
int ch,id,f=1;
struct user *temp;
time_t logout=time(NULL);
printf("\n\t\t\t\t %d  is the login time && %d is the logiut time",login,logout);
double diff=difftime(logout,login);
printf("%.2f is the time used\t\t\t\t",diff);
int bill=(int)(diff *50);
printf("\n\t\t\t\t\t Your bill is %d",bill);
printf("\n\t\t\t\t\tPlease choose the method to pay the bill");
printf("\n\t\t\t\t\t 1.IC card\n\t\t\t\t\t2.Spot Cash");
scanf("%d",&ch);
if(ch==1)
        printf("\n\t\t\t\tplease enter the user_id to continue");
        scanf("%d",&id);
        if(id==uid)
        for(temp=first;temp!=NULL;temp=temp->next)
        {
                if(temp->user_id==id)
                {
                        printf("\n\t\t\t\t\t %d amount have been deducted from your
account \t",bill);
                        temp->balance-=bill;
                        break;
                }
                else
                f=0;
        }
        if(f==0)
        printf("\n\t\t\t\t\t sorry you dont have an account in our cafe");
}



void  del()
{       char ch[3];
int x,loc=0,id;
                struct user *temp=first;

printf("\n\t\t\t\t THE RECORDS OF OUR USERS ARE:\n");
printf("\n\t\t\t\t\tUsername:\t\t UserId");
        for(temp=first;temp!=NULL;temp=temp->next)
        {
        printf("\n\t\t\t\t-------------------------------------------------------------\n");
        printf("\n\t\t\t\t\t%s\t\t%d",temp->name,temp->user_id);
        }
printf("\n\t\t\t\t-------------------------------------------------------------\n");
printf("\n\t\t\t\t\tPlease enter the user id\t");
```

```
scanf("%d",&id);
temp=first;
        while(temp!=NULL)
        {       loc++;
                if(temp->user_id==id)
                {
                        printf("\n\t\t\t\t\t\t %d is your number \t",loc);
                        DelUser(loc);
                }
                temp=temp->next;
        }
        if(temp==NULL)
                printf("\n\t\t\t\t\t User doesnt exist in our record");
getch();
system("cls");
}

 void change()
 {
        char name[122],passw[33];
        int x;

        printf("\n\t\t\t\t\t\t PLEASE CONFIRM YOU ARE ADMIN ");
        printf("\n\t\t\t\t\t\t ENTER THE PREVIOUS CREDENTIALS");
        printf("\n\t\t\t\t\t\tUSER NAME :\t");
        scanf("%s",name);
        printf("\n\t\t\t\t\t\t PASSWORD:\t");
        scanf("%s",passw);
        x=strcmp(name,owner);
        if(x==0)
 {
                x=strcmp(passw,pass);
                if(x==0)
```

```
            {
                        printf("\n\t\t\t\t\t\t ENTER YOUR NEW USERNAME\t");
                        scanf("%s",ad->user_name);
                        printf("\n\t\t\t\t\t\t ENTER THE PASSWORD\t");
                        scanf("%s",ad->password);
            }
 }
printf("\n\t\t\t\t\tPress any key to continue..");
getch();
system("cls");
admin();
 }

 void history()
 {
            char ch;
            printf("\n\t THE HISTORY OF PAST SESSIONS IS");
            FILE *fptr=fopen("Internet.txt","r");
            while(fscanf(fptr,"%c",&ch)!=EOF)
            {
                        if(ch=='\n')
             printf("\n");
            else
            {
                        printf("%c",ch);
            }
}
fclose(fptr);
 }

 void feedback()
 {          int id;
            struct user *temp =first;
```

```
        printf("\n\t\t\t\t\t\t ENTER THE USER ID TO CONTINUE\t");
        scanf("%d",&id);
        while(temp!=NULL)
        {
                if(temp->user_id==id)

         {
                printf("\n\t\t\t\t\t\t YOUR RESPOINSE WILL HELP US A LOT ");
                        printf("\n\t\t\t\t\t\t PLEASE ENTER YOUR FEEDBACK\t");
                        fflush(stdin);
                        gets(temp->feedback);
         }
         temp=temp->next;
}
printf("\n\t\t\t\t\tPress any key to continue..");
getch();
system("cls");
free(temp);
 }

 void review()
{
        printf(" \n\t\t\t\t\t\tDear Admin,Here are the feedbacks by your beloved
users\t");
  struct user *temp;
  printf("\n\t Name\t\tFeedback\t\tUser Id\t\tRegistration time\t");
  for(temp=first;temp!=NULL;temp=temp->next)
  {
printf("\n-------------------------------------------------------------------------------------------------
----\n");

        printf(" \t%s\t\t%s\t\t\t%d\t%d\t\t%s",temp->name,temp->feedback,temp-
>user_id,temp->balance,temp->reg_time);
```

```
  }
          printf("\n------------------------------------------------------------------------------------------
-------------\n");
                    printf("Enter key to continue");
                    getch();
          system("cls");
          free(temp);

}

void DelUser(int loc)
{
static int a=1000;
struct user *temp,*temp1;

if(first== NULL)
          printf("n\t\t\t\t\t\t\t\t\t\t NO USERS ");
if(loc==1)
{
          temp=first;
          temp=temp->next;
          first=temp;

}
else
{
          temp=first;
          if(temp->next==NULL)
                    first =NULL;
          else
          {
```

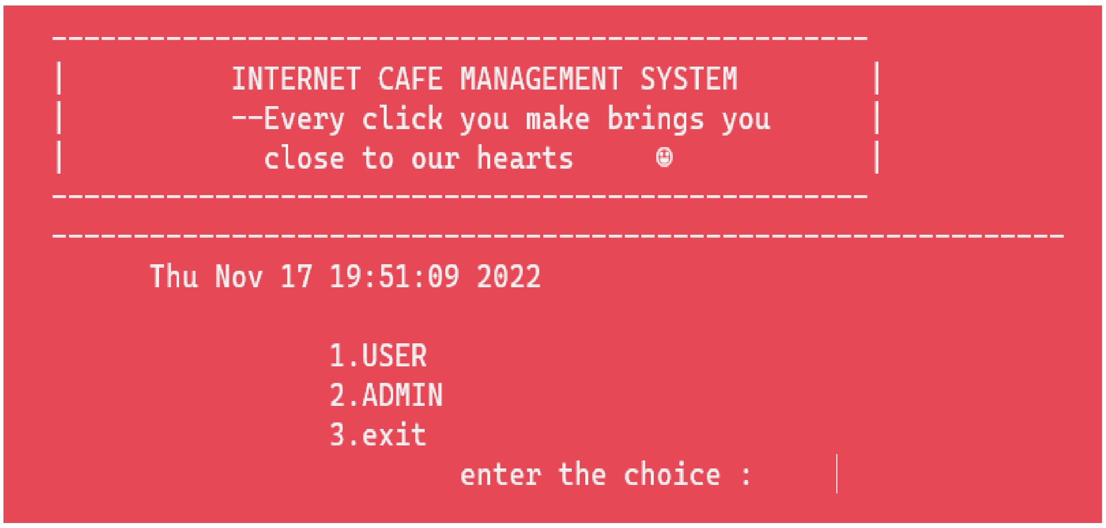
```
                    while(--loc)
          {
                    temp1=temp;
                    temp=temp->next;
          }
          temp1->next=temp->next;
for(temp=first;temp!=NULL;temp=temp->next)
          temp->user_id=++a;
printf("\n\t\t\t\t\t\t  YOUR ACCOUNT HAS BEEN DELETED");
          }
}
printf("\n\t\t\t\t\t Press any key to continue..");
getch();
system("cls");
admin();

}

int log_in()
{
int user_id,ch,cmp,flag=0;
char password[MAX_SIZE];
          if(first==NULL)
{
                    printf("\n\t\t\t\t\t\t Our Record is empty please be our valuable
customer");
                    printf("\n\t\t\t\t\t\t Enter 1 to create an account\n\t\t\t\t\t\tEnter 2 to
exit \n\t\t\t\t\t\t");
                    scanf("%d",&ch);
                    if(ch==1)
                    {
                              reg();
                    }
                    else
                    printf("n\t\t\t\t\t\t\t\t\t\t\t Thanks for visiting the Cafe");
}
else
{

printf("\n\t\t\t\t\t\t Enter the user id:\t");
scanf("%d",&user_id);
struct user *temp,*temp2;
for(temp=first;temp!=NULL;temp=temp->next)
{
          if( user_id==temp->user_id)
          {
                    flag=1;
                    temp2=temp;
                    break;
          }
          else
          {
                    flag=0;
          }
}
          if(flag==1)
 {
             fflush(stdin);
             a:
             printf("\n\t\t\t\t\t\t Enter the password\t");
                    scanf("%s",password);
                    cmp=strcmp(password,temp2->userpass);
                    if(cmp==0)
                    {
                              printf("\n\t\t\t\t\t\t Successful Login\n");
                              time_t login=time(NULL);
```

```
                        printf("\n\t\t\t\t\t\t Enter 1 to go to bill");
                        while(1)
                        {
                                scanf("%d",&ch);
                                if(ch==1)
                                {
                                        bill(login,temp->user_id);
                                        break;
                                }

                        }
                }
                else
                {
                        printf("\n\t\t\t\t\t\t Please check your credentials and retry 
again");
                        goto a;
                }
        }
        else if(flag==0)
        {
                printf("\n\t\t\t\t\t\t Sorry Dear ! you donot have an account here\n");
                printf("\n\n Enter 1 to create an account\n\t\t\t\t\t\tEnter 2 to exit 
\n");
                scanf("%d",&ch);
                if(ch==1)
                reg();
                else
                printf("\n\t\t\t\t\t\t Thanks for visiting the Cafe");
        }
}
          return 0;
```

```
                system("cls");
}

void file_update()
{
int i=1;
struct user *temp;
time_t now=time(NULL);
char *st=ctime(&now);
FILE *fptr=fopen("Internet.txt","a+");
fprintf(fptr,"\n\tDate: %s",st);
fprintf(fptr,"\nS.No   Name \tID\tFeedback\t\tBalance\t\tRegistration Time");
for(temp=first;temp!=NULL;temp=temp->next)
{
        fprintf(fptr,"\n  %d   %s \t %d\t%s\t%d \t%s",i++,temp->name,temp-
>user_id,temp->feedback,temp->balance,temp->reg_time);
}
fprintf(fptr,"\n-------------------------------------------------------------------------------------");
fclose(fptr);
}
```

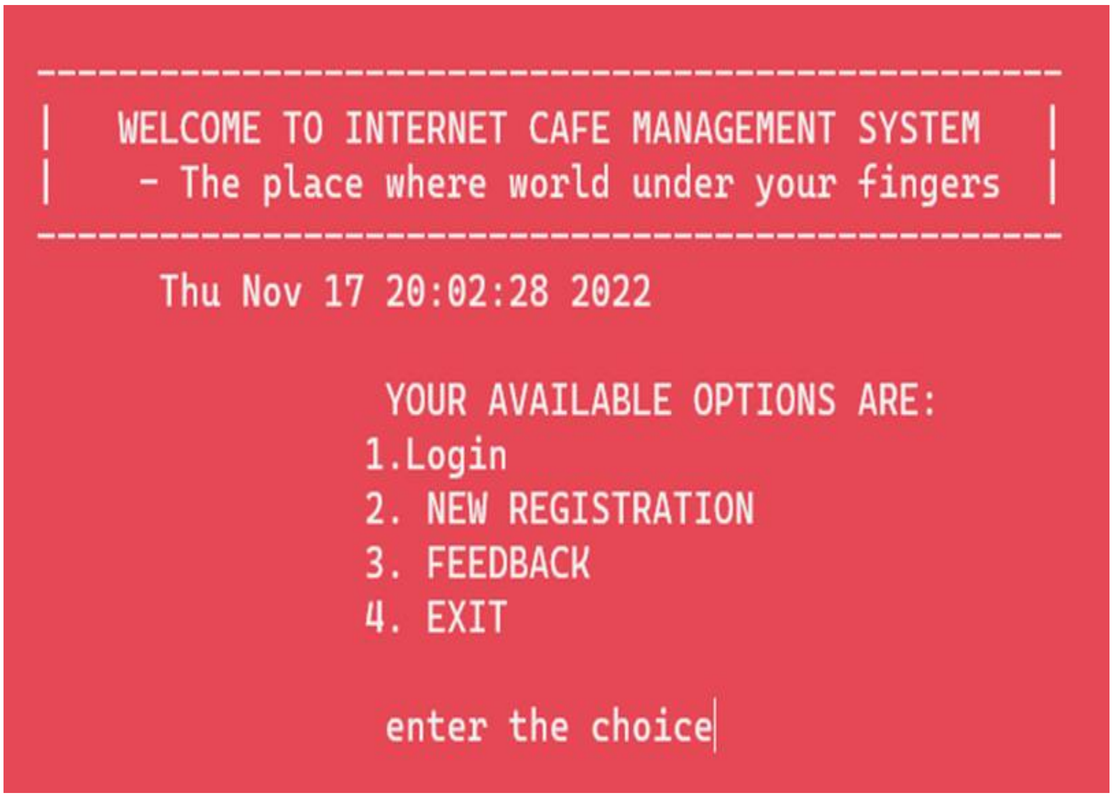
OUTPUT :

Main Menu:



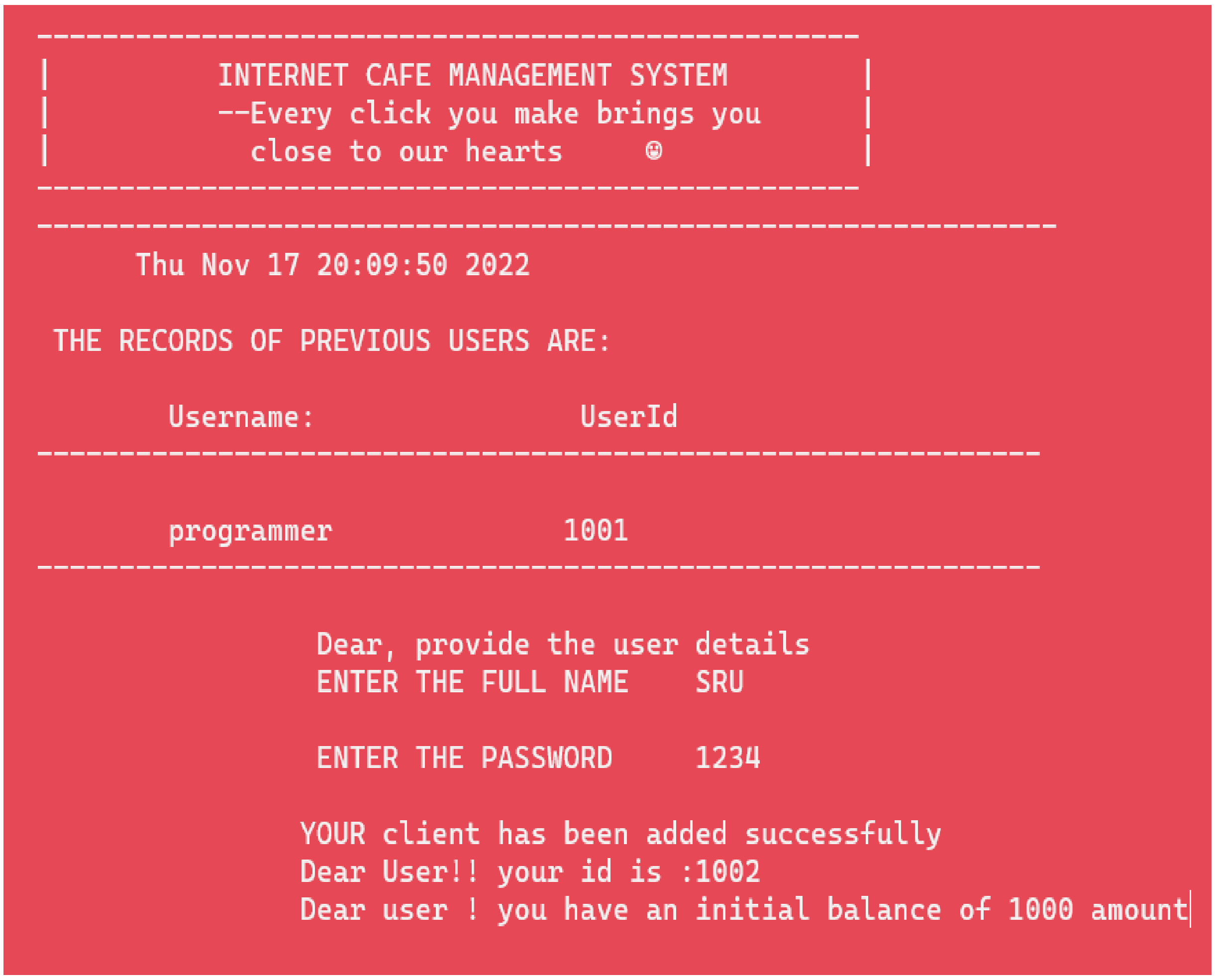
```
 --------------------------------------------------
 |          INTERNET CAFE MANAGEMENT SYSTEM         |
 |          --Every click you make brings you       |
 |            close to our hearts     ☻             |
 --------------------------------------------------
 --------------------------------------------------------------
       Thu Nov 17 19:51:09 2022

                  1.USER
                  2.ADMIN
                  3.exit
                          enter the choice :
```

User Options:



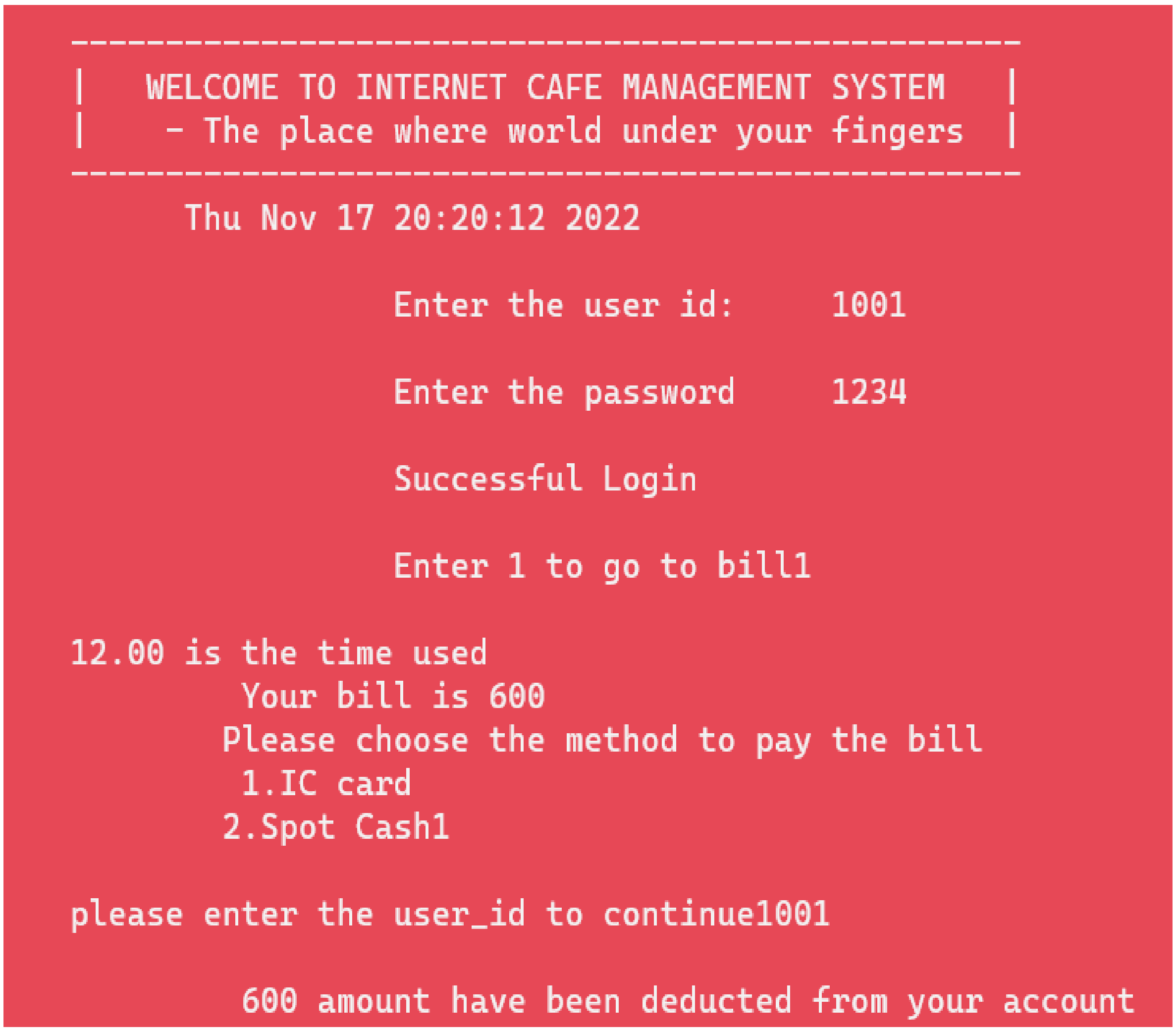
```
-----------------------------------------------------
|   WELCOME TO INTERNET CAFE MANAGEMENT SYSTEM      |
|    - The place where world under your fingers     |
-----------------------------------------------------
       Thu Nov 17 20:02:28 2022

                   YOUR AVAILABLE OPTIONS ARE:
                  1.Login
                  2. NEW REGISTRATION
                  3. FEEDBACK
                  4. EXIT

                   enter the choice
```

Registration :



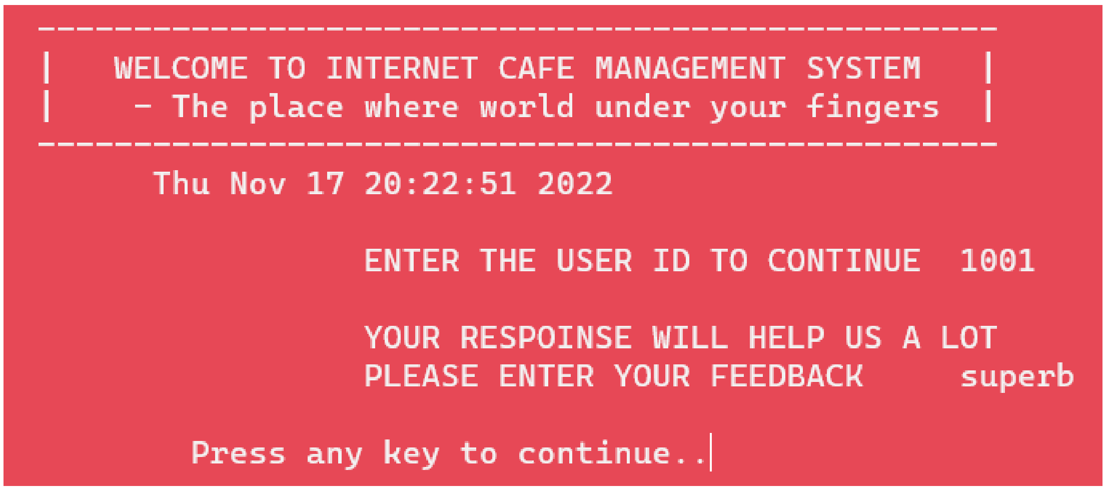
```
---------------------------------------------------
|          INTERNET CAFE MANAGEMENT SYSTEM         |
|          --Every click you make brings you       |
|            close to our hearts     ☻             |
---------------------------------------------------
---------------------------------------------------------------
      Thu Nov 17 20:09:50 2022

 THE RECORDS OF PREVIOUS USERS ARE:

        Username:                 UserId
--------------------------------------------------------------

        programmer               1001
--------------------------------------------------------------

                 Dear, provide the user details
                 ENTER THE FULL NAME    SRU

                 ENTER THE PASSWORD     1234

                YOUR client has been added successfully
                Dear User!! your id is :1002
                Dear user ! you have an initial balance of 1000 amount
```

Login & Bill payment:

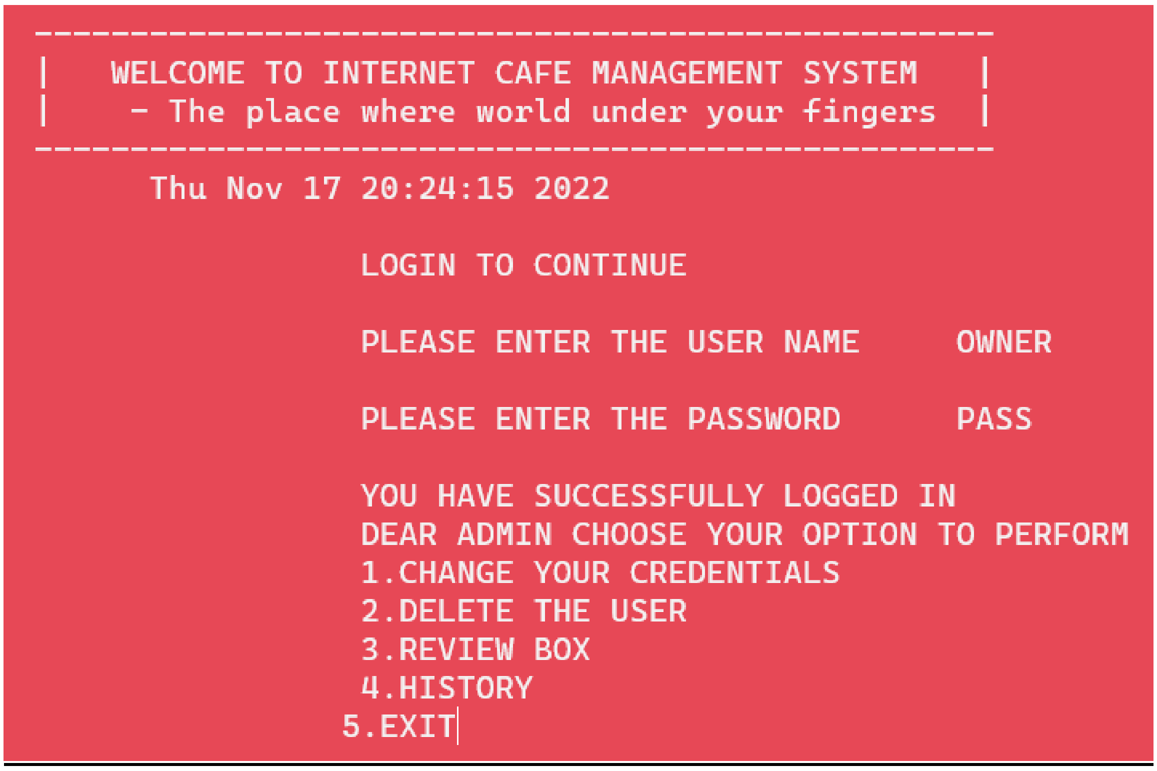


```
-------------------------------------------------
|   WELCOME TO INTERNET CAFE MANAGEMENT SYSTEM   |
|    - The place where world under your fingers  |
-------------------------------------------------
      Thu Nov 17 20:20:12 2022

                 Enter the user id:     1001

                 Enter the password     1234

                 Successful Login

                 Enter 1 to go to bill1

12.00 is the time used
         Your bill is 600
        Please choose the method to pay the bill
         1.IC card
        2.Spot Cash1

please enter the user_id to continue1001

         600 amount have been deducted from your account
```

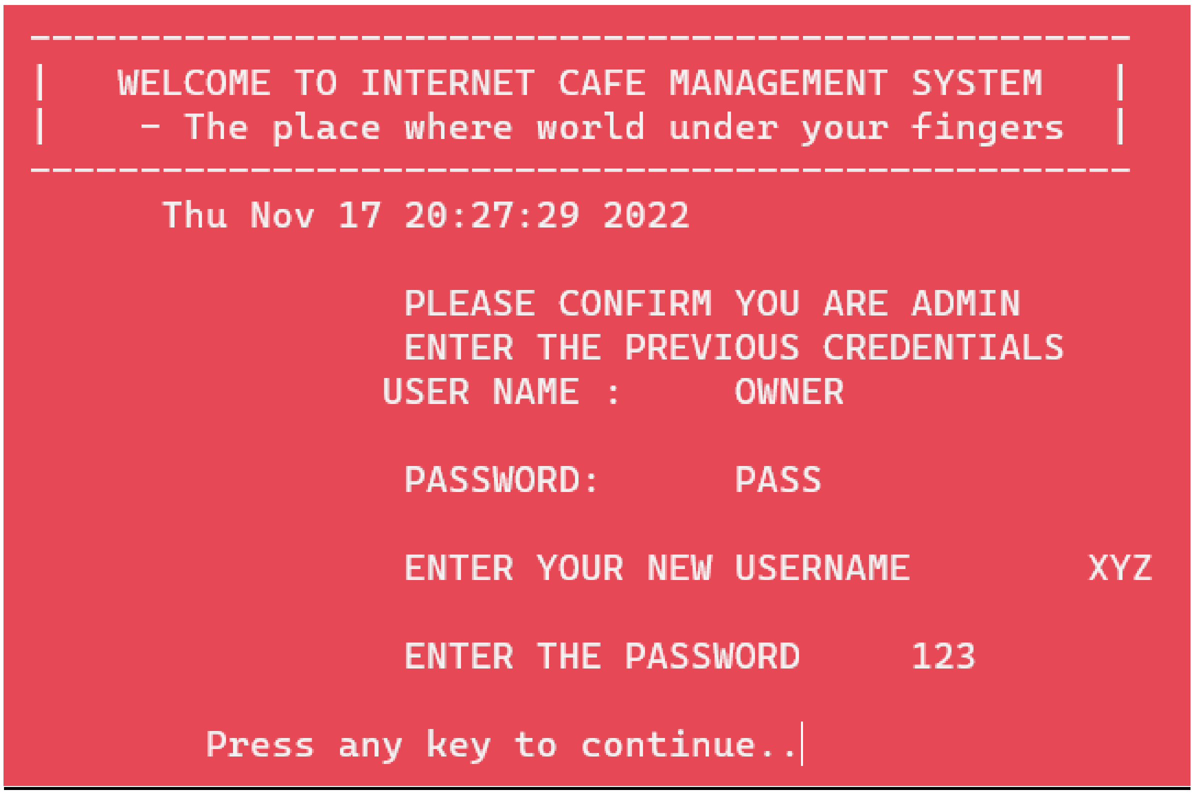
Feedback:


```
 -------------------------------------------------
 |   WELCOME TO INTERNET CAFE MANAGEMENT SYSTEM   |
 |    - The place where world under your fingers  |
 -------------------------------------------------
       Thu Nov 17 20:22:51 2022

                  ENTER THE USER ID TO CONTINUE  1001

                  YOUR RESPOINSE WILL HELP US A LOT
                  PLEASE ENTER YOUR FEEDBACK     superb

         Press any key to continue..
```

Admin options:

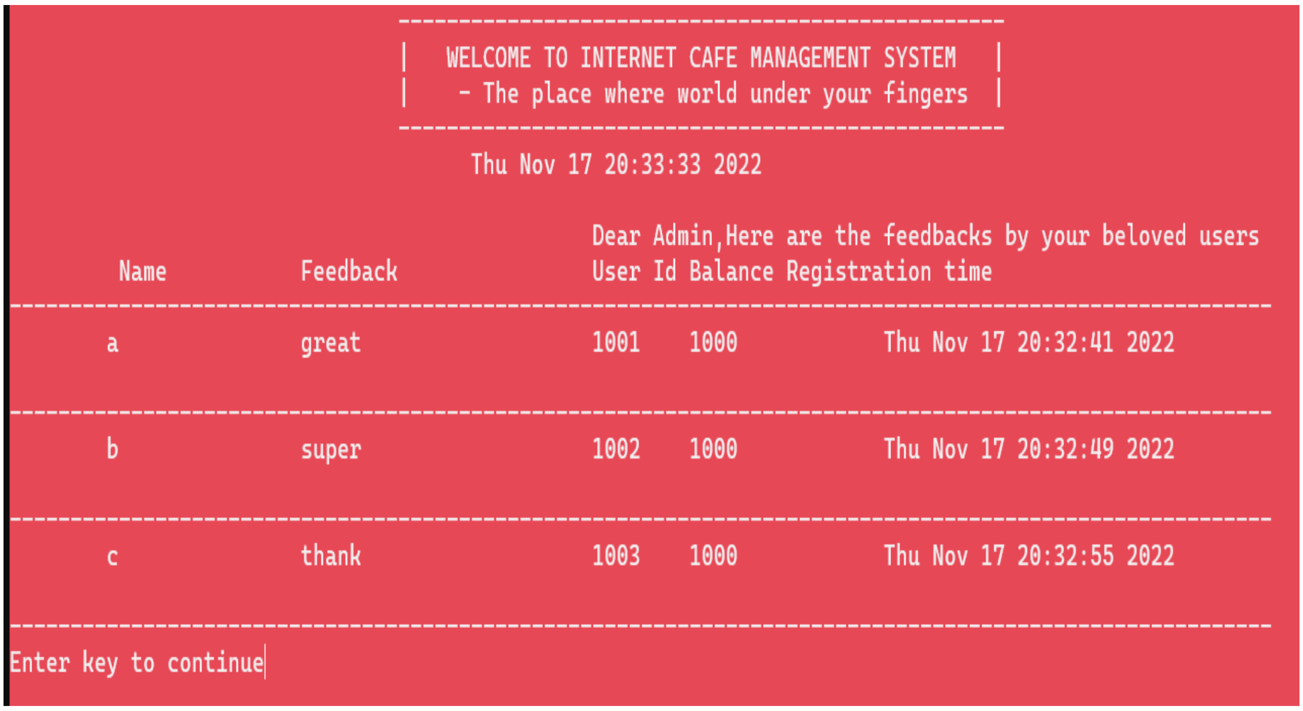

```
 -------------------------------------------------
 |   WELCOME TO INTERNET CAFE MANAGEMENT SYSTEM   |
 |    - The place where world under your fingers  |
 -------------------------------------------------
       Thu Nov 17 20:24:15 2022

                  LOGIN TO CONTINUE

                  PLEASE ENTER THE USER NAME     OWNER

                  PLEASE ENTER THE PASSWORD      PASS

                  YOU HAVE SUCCESSFULLY LOGGED IN
                  DEAR ADMIN CHOOSE YOUR OPTION TO PERFORM
                  1.CHANGE YOUR CREDENTIALS
                  2.DELETE THE USER
                  3.REVIEW BOX
                  4.HISTORY
                 5.EXIT
```

Credentials updating page:



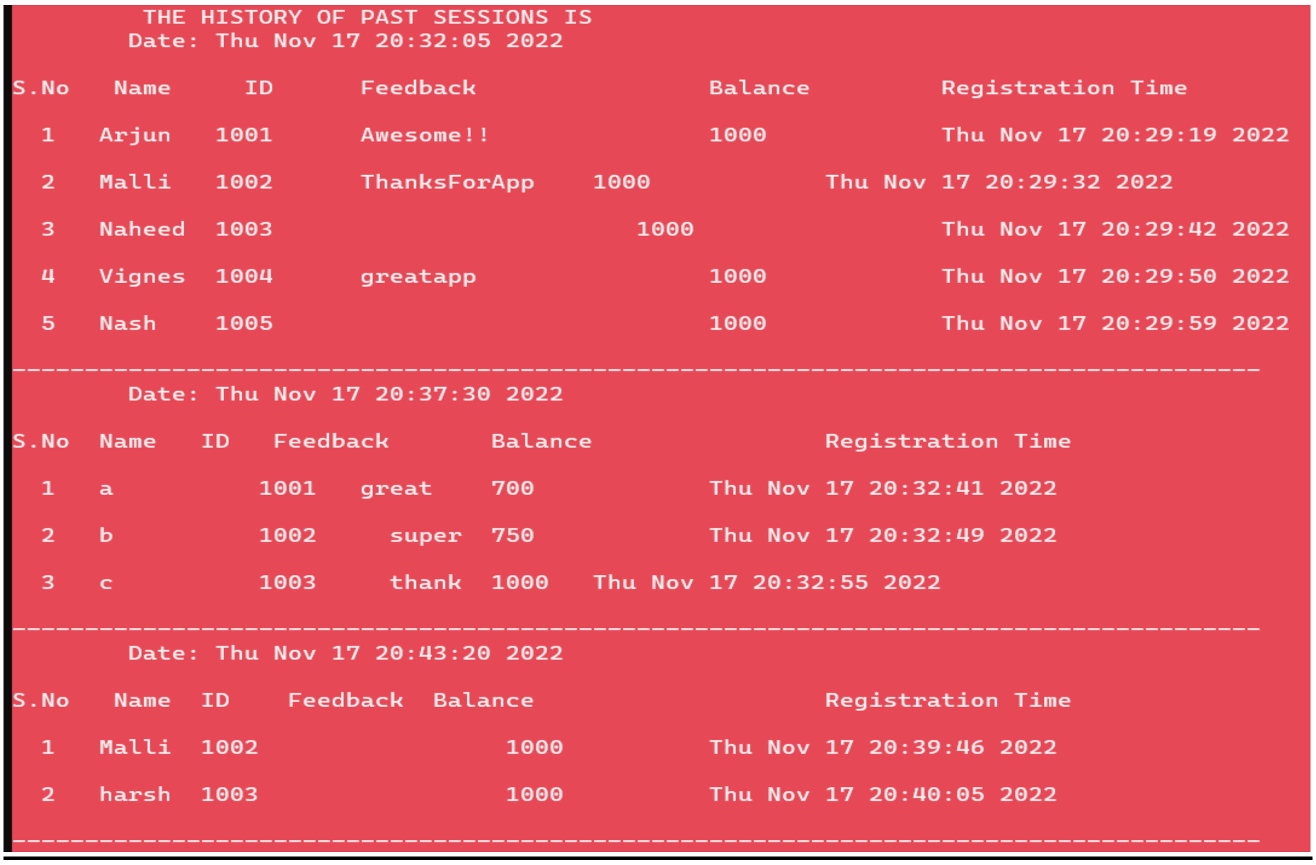
```
 ---------------------------------------------------
|   WELCOME TO INTERNET CAFE MANAGEMENT SYSTEM    |
|    - The place where world under your fingers   |
 ---------------------------------------------------
      Thu Nov 17 20:27:29 2022

                  PLEASE CONFIRM YOU ARE ADMIN
                  ENTER THE PREVIOUS CREDENTIALS
                 USER NAME :     OWNER

                  PASSWORD:      PASS

                  ENTER YOUR NEW USERNAME        XYZ

                  ENTER THE PASSWORD     123

         Press any key to continue..
```

Reviews:



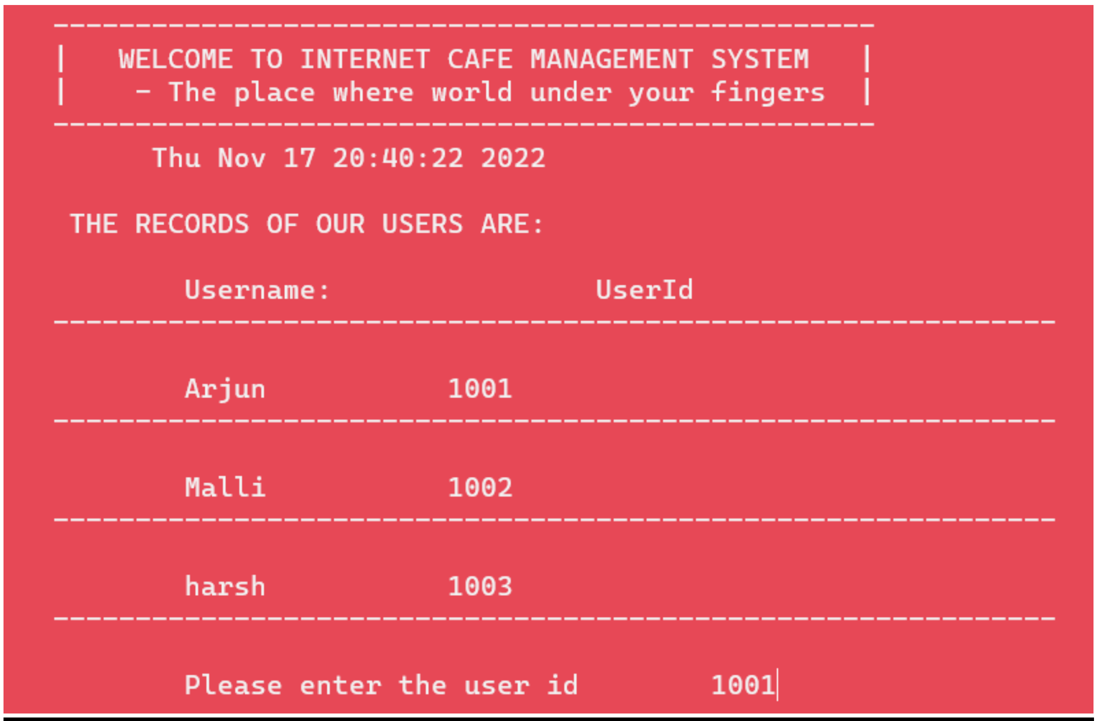
```
                          ---------------------------------------------------
                          |   WELCOME TO INTERNET CAFE MANAGEMENT SYSTEM   |
                          |    - The place where world under your fingers  |
                          ---------------------------------------------------
                               Thu Nov 17 20:33:33 2022

                                          Dear Admin,Here are the feedbacks by your beloved users
      Name          Feedback              User Id Balance Registration time
-----------------------------------------------------------------------------------------------
      a             great                 1001    1000          Thu Nov 17 20:32:41 2022

-----------------------------------------------------------------------------------------------
      b             super                 1002    1000          Thu Nov 17 20:32:49 2022

-----------------------------------------------------------------------------------------------
      c             thank                 1003    1000          Thu Nov 17 20:32:55 2022

-----------------------------------------------------------------------------------------------
Enter key to continue
```

History of last sessions:



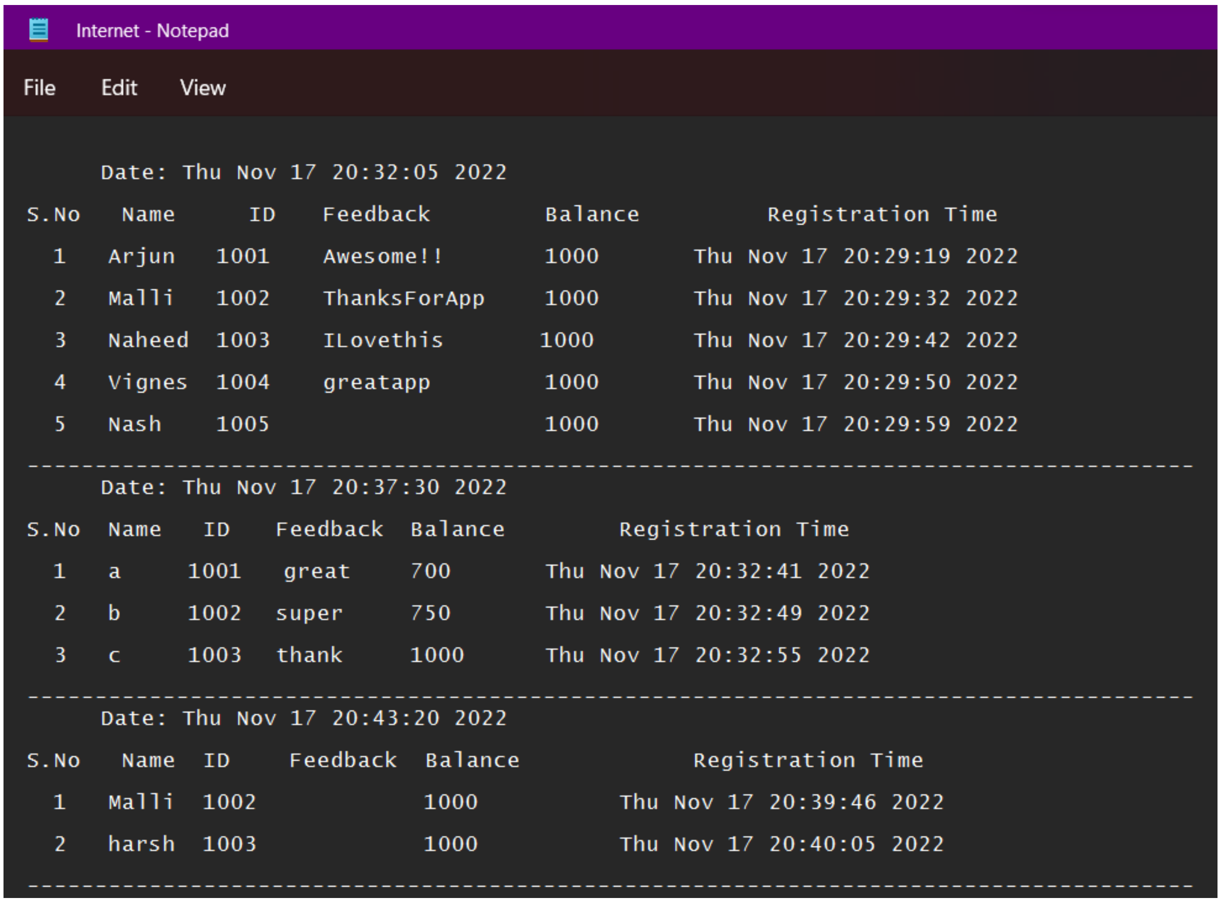
```
         THE HISTORY OF PAST SESSIONS IS
        Date: Thu Nov 17 20:32:05 2022

S.No   Name     ID      Feedback                Balance          Registration Time

  1   Arjun   1001      Awesome!!               1000             Thu Nov 17 20:29:19 2022

  2   Malli   1002      ThanksForApp    1000            Thu Nov 17 20:29:32 2022

  3   Naheed  1003                         1000                  Thu Nov 17 20:29:42 2022

  4   Vignes  1004      greatapp                1000             Thu Nov 17 20:29:50 2022

  5   Nash    1005                              1000             Thu Nov 17 20:29:59 2022

--------------------------------------------------------------------------------------
        Date: Thu Nov 17 20:37:30 2022

S.No  Name   ID   Feedback       Balance                Registration Time

  1   a          1001   great    700            Thu Nov 17 20:32:41 2022

  2   b          1002     super  750            Thu Nov 17 20:32:49 2022

  3   c          1003     thank  1000   Thu Nov 17 20:32:55 2022

--------------------------------------------------------------------------------------
        Date: Thu Nov 17 20:43:20 2022

S.No   Name  ID    Feedback  Balance                    Registration Time

  1   Malli  1002                1000          Thu Nov 17 20:39:46 2022

  2   harsh  1003                1000          Thu Nov 17 20:40:05 2022

--------------------------------------------------------------------------------------
```

Deletion of user:



```
 -------------------------------------------------
 |   WELCOME TO INTERNET CAFE MANAGEMENT SYSTEM   |
 |    - The place where world under your fingers  |
 -------------------------------------------------
       Thu Nov 17 20:40:22 2022

  THE RECORDS OF OUR USERS ARE:

        Username:                    UserId
 -------------------------------------------------------------


        Arjun           1001
 -------------------------------------------------------------


        Malli           1002
 -------------------------------------------------------------


        harsh           1003
 -------------------------------------------------------------

        Please enter the user id        1001
```

Record of the internet cafe:



```
     Date: Thu Nov 17 20:32:05 2022
S.No   Name     ID    Feedback         Balance         Registration Time
  1   Arjun    1001    Awesome!!        1000       Thu Nov 17 20:29:19 2022
  2   Malli    1002    ThanksForApp     1000       Thu Nov 17 20:29:32 2022
  3   Naheed   1003    ILovethis       1000        Thu Nov 17 20:29:42 2022
  4   Vignes   1004    greatapp         1000       Thu Nov 17 20:29:50 2022
  5   Nash     1005                     1000       Thu Nov 17 20:29:59 2022
-------------------------------------------------------------------------------------
     Date: Thu Nov 17 20:37:30 2022
S.No  Name   ID    Feedback  Balance         Registration Time
  1   a     1001    great     700       Thu Nov 17 20:32:41 2022
  2   b     1002   super     750       Thu Nov 17 20:32:49 2022
  3   c     1003   thank     1000      Thu Nov 17 20:32:55 2022
-------------------------------------------------------------------------------------
     Date: Thu Nov 17 20:43:20 2022
S.No   Name  ID    Feedback  Balance            Registration Time
  1   Malli  1002            1000          Thu Nov 17 20:39:46 2022
  2   harsh  1003            1000          Thu Nov 17 20:40:05 2022
-------------------------------------------------------------------------------------
```

Conclusion

From this project I want to conclude that the knowledge we gained by learning the Data Structures Laboratory has been applied successfully and the project can be implemented in any system that meet the specified requirements.

- This application can also be extended to implement in other environments like Library, Hotels etc.
- In future changes can be done by linking the user mobile number or Aadhar card or their fingerprints and allowed to modify their data whenever needed.

References

- Data Structures (Linked List) in Geeks for Geeks.
- Problems in most of the Internet cafes (storing user data).