DSA [20ES117] COURSE PROJECT REPORT

On

"METRO TICKET BOOKING SYSTEM"

Developed By:

H.T.NO	-	2103A54008
STUDENT NAME	-	P.AJAY GOUD

Under the Guidance

of

Dr. A. SIVA KRISHNA REDDY

Associate Professor

Submitted to

Department of 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 **"METRO TICKET BOOKING SYSTEM** is a record of bona fide work carried out by the student **"Pochanagari Ajay Goud"** bearing roll number **"2103A54008"** of Computer Science and Artificial Intelligence department during the academic year 2022-23.

Supervisor	**Head of the Department**
(Dr. A. Siva Krishna Reddy)	(H.O.D)

INDEX

IDEA

- My idea is to provide an application that helps people to by tickets without waiting in a line and wasting their time.
- Easily they can create their cards and delete cards.
- The information of person with name and mobile number with their tickets.
- It is implemented using Files and Data Structure – Singly Linked List.

PROBLEM STATEMENT:

- People don’t have proper access to get tickets .
- Ticket agent doesn’t contain the people information for further notifications.

Develop a C program to store details of bookings of tickets. The application also provides the following mentioned functionalities:

There are five options

1.view Stations

2. Buy tickets

3.create card

4.delete card

5.Travellers History

MODULES:

In this application many of the 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 in this program is done dynamically.

In this application, the following modules are used:

1. **void green()**
   Selection of green line stations that are available from 1 to 10 stations

Station names:- MG Bus, Sultan Bazar, Narayanguda, Chikkadpally, RTCXRoads, Musheerabad, Gandhi Hospital, Secunderabad West, Parade Ground, JBS Parade Ground.

2. **void blue()**

   Nagole, Uppal, Stadium,N GRI, Habsiguda,Tarnaka, Mettuguda,Seunderabad East, Paradise, Rasoolpura,Prakashnagar,Begumpet,Ameerpet, Madhura Nagar, Yusfguda Jubille hills, Jubille hills check post, Peddama gudi, Madhapur.Durgam Cheruvu, Hitec City.

3. **void red()**
   LB Nagar, Victoria Memorial, ChaitanyaPuri, Dilsukhnagar, Musarambagh, New market, Malakpet, MG BUS, Osmaina Medical College, Gandhi Bhavan, Nampally,Assembly, Lakdhi-ka-pul, Khairtabad, Irrum manzil, Panja Gutta,Ameerpet S R Nagar,Esi Hospital, Erragadda, Bharat Nagar,Moosapet, Balanagar, Kukatpally, Kphb colony, JNTU,Miyapur.

4. **void stations();**
   To display the all the station lines.
5. **void create_card();**
   We can create our card by giving additional informations like name mobile number,email id and it will generate the password through our name and mobile number.
6. **void delete_card();**
   We can delete our existings cards.
7. **void time1();**
   For every 5 minutes train is available on the track so ticket shows 5 minutes to arrive train from any Station.

USAGE OF TOOLS

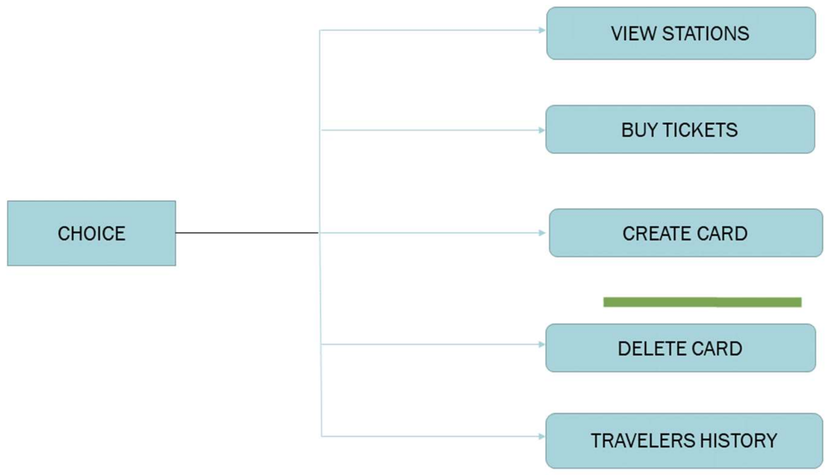
Hardware Tools:

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
- Strings (strings) and its functions (strcmp)
- Functions (Any type of user defined functions)
- Structures
- Pointers (pointers to structures)
- Dynamic Memory Allocation (malloc/ calloc/ realloc)
- Linked Lists (Single Linked List)
- Windows library functions

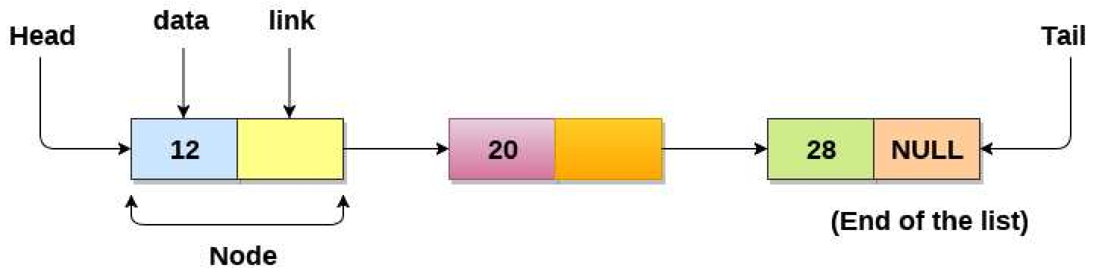
SOLUTION

- The application can be implemented in order to manage its customers and their information.
- Firstly, the interface takes the information of ticket history and initial username and mobile number.
- We can create our own card by providing name,mobile number and email id.
- We can delete our existings card which is not work.
- We can see the travelers history by their mobile numbers.

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.

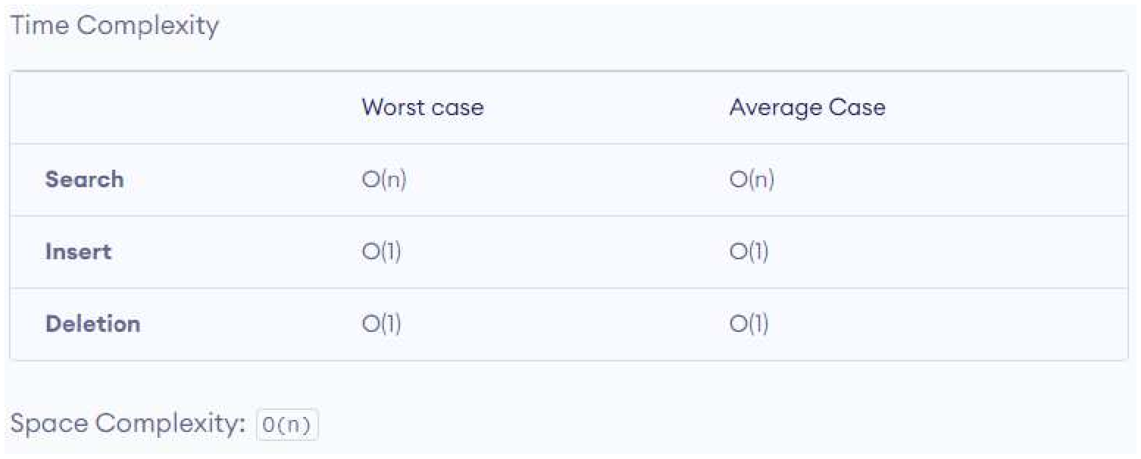
Array contains following limitations:

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 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:

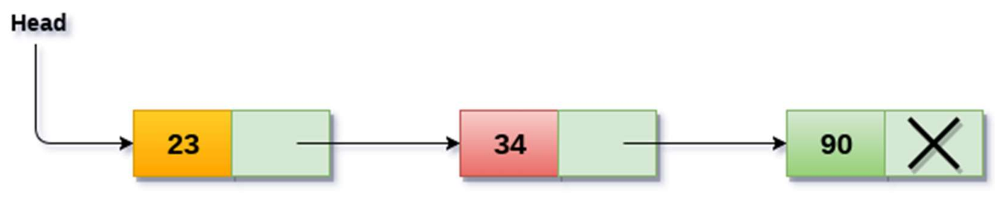
Here's a list of basic linked list operations that we will cover in this article.

- 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.

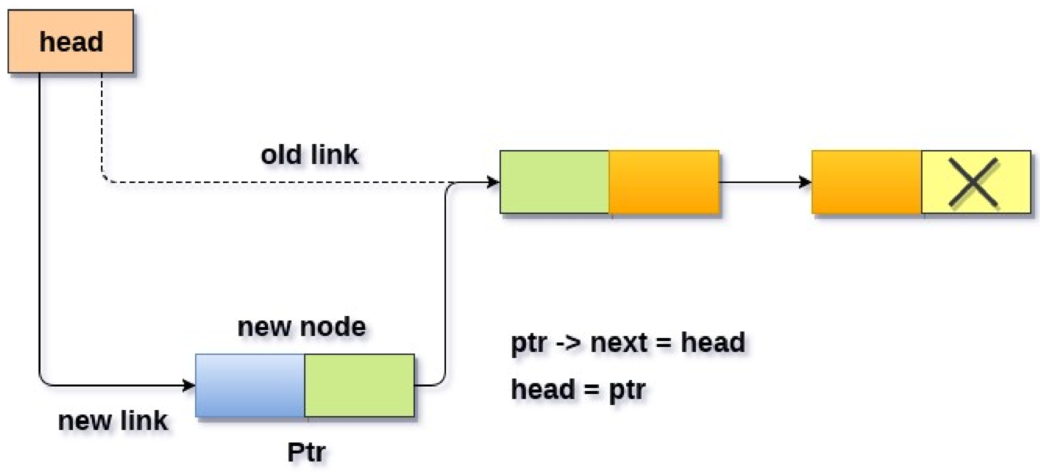
1. ptr = (struct node *) malloc(sizeof(struct node *));
2. ptr → data = item

- 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.

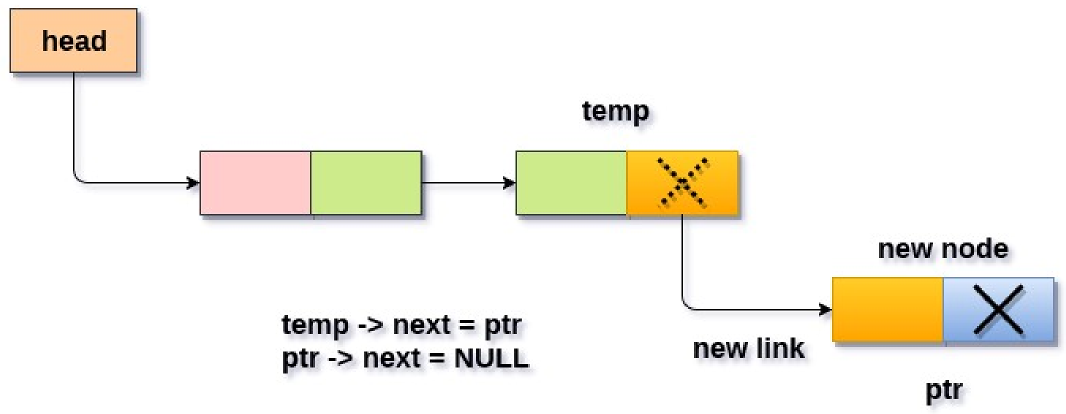
1. Temp = head

- 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.

```
1. temp=head;
2.          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.

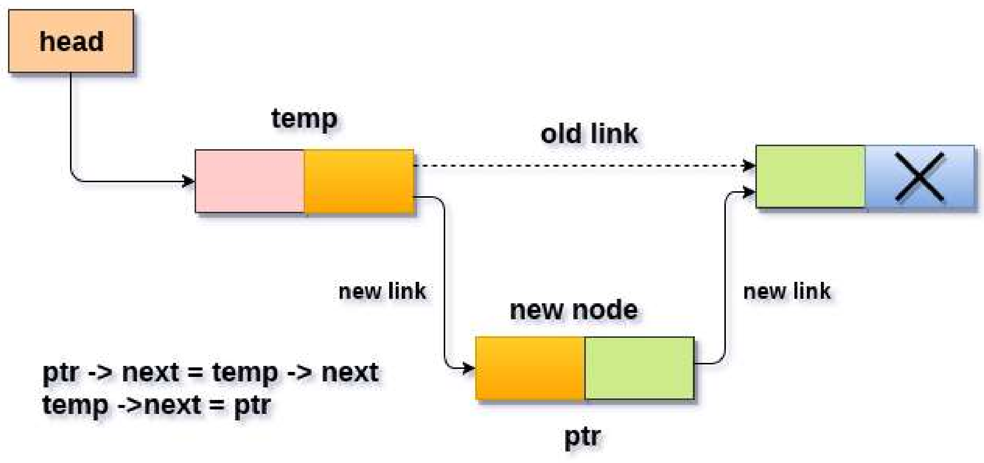
```
ptr = (struct node *) malloc (sizeof(struct node));
        ptr->data = item;
```

- 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:

```
o STEP 1: IF PTR = NULL
WRITE OVERFLOW
   GOTO STEP 12
  END OF IF
o STEP 2: SET NEW_NODE = PTR
o STEP 3: NEW_NODE → DATA = VAL
o STEP 4: SET TEMP = HEAD
o STEP 5: SET I = 0
o STEP 6: REPEAT STEP 5 AND 6 UNTIL I<loc< li=""></loc<>
o STEP 7: TEMP = TEMP → NEXT
o STEP 8: IF TEMP = NULL
WRITE "DESIRED NODE NOT PRESENT"
    GOTO STEP 12
   END OF IF
 END OF LOOP
o STEP 9: PTR → NEXT = TEMP → NEXT
o STEP 10: TEMP → NEXT = PTR
o STEP 11: SET PTR = NEW_NODE
o STEP 12: EXIT
```

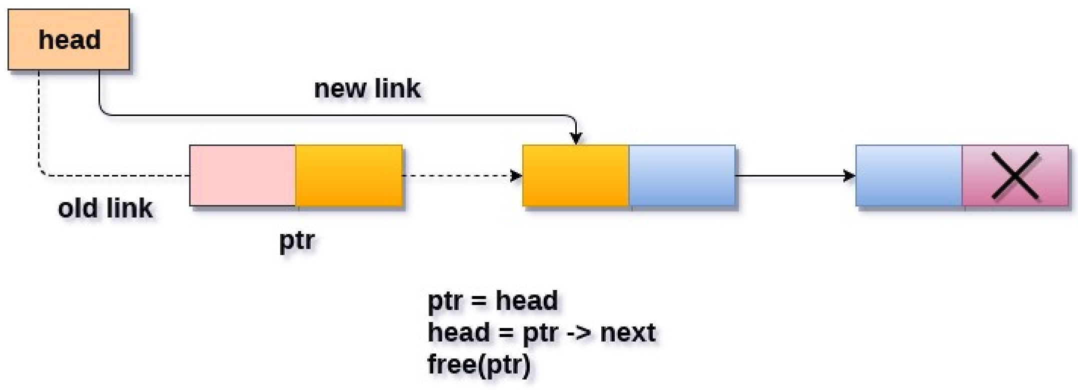
Deletion of Element in Single Linked List at Beginning:

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.

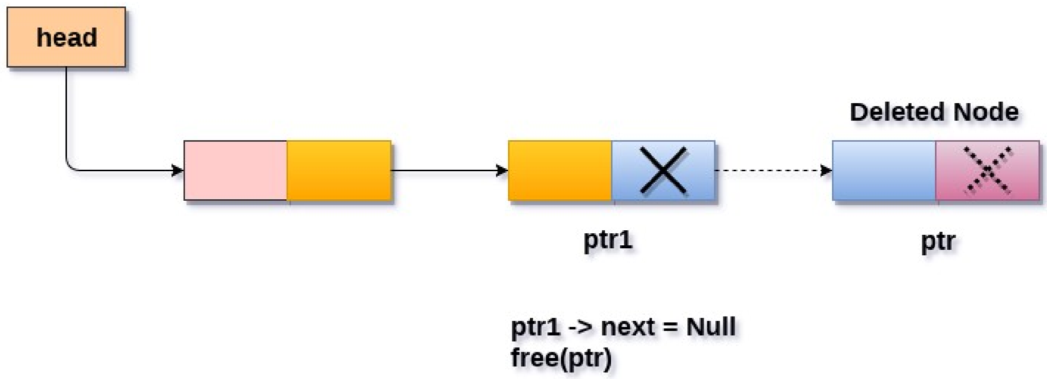
For this purpose, just declare a temporary pointer temp and assign it to head of the list. We also need to keep track of the second last node of the list. For this purpose, two pointers ptr and ptr1 will be used where ptr will point to the last node and ptr1 will point to the second last node of the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

```
o Step 1: IF HEAD = NULL

Write UNDERFLOW
  Go to Step 8
 [END OF IF]

o Step 2: SET PTR = HEAD

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

o Step 4: SET PREPTR = PTR

o Step 5: SET PTR = PTR -> NEXT

[END OF LOOP]

o Step 6: SET PREPTR -> NEXT = NULL

o Step 7: FREE PTR

o 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.

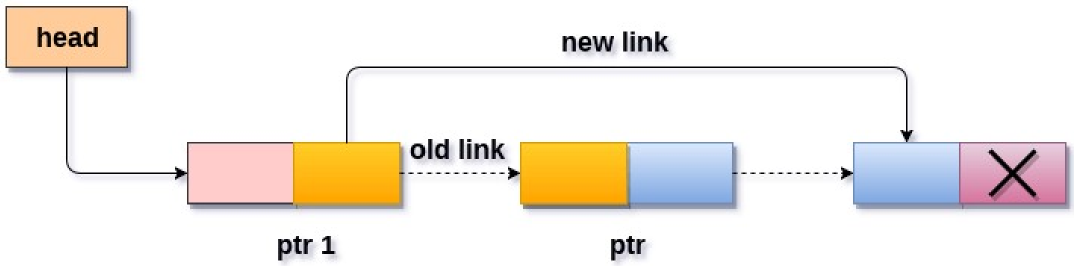
Use the following statements to do so.

```
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);
10.              return;
11.           }
12.       }
```

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.

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



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

Algorithm:

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

WRITE UNDERFLOW
GOTO STEP 10
END OF IF

- **STEP 2:** SET TEMP = HEAD
- **STEP 3:** SET I = 0
- **STEP 4:** REPEAT STEP 5 TO 8 UNTIL I<loc< li=""></loc<>
- **STEP 5:** TEMP1 = TEMP
- **STEP 6:** TEMP = TEMP → NEXT
- **STEP 7:** IF TEMP = NULL

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

- **STEP 8:** I = I+1

END OF LOOP

- **STEP 9:** TEMP1 → NEXT = TEMP → NEXT
- **STEP 10:** FREE TEMP
- **STEP 11:** EXIT

SOURCE CODE:

```
/*
   DEVLOPER:-AJAY*/
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int count=0;
int no;
```

```
struct node
{
        char data[20],number[11];
        struct node *link;
}*start=NULL,*newnode,*temp,*p;



typedef struct {
        char num[5];
        char name[100];
        int age;
        char number[20];
        int balance;
} card;

void stations();
void green();
void blue();
void red();
void green1();
void blue1();
void red1();
int ajay(int);
void buy();
void morebuy();
void create_card();
void delete_card();
void ajaycash();
void pay_cash();
void pay_card();
void time1();
void colors();



void green()
{
        green1();
        colors();
}
void blue()
{
   blue1();
         colors();
}
void red()
{
        red1();
        colors();
}
void green1()
{
        system("color 2");
        printf("\n1.MG Bus\n2.Sultan Bazar\n3.Narayanguda\n4.Chikkadpally\n5.RTC X
Roads\n");
        printf("6.Musheerabad\n7.Gandhi Hospital\n8.Secunderabad West\n9.Parade
Ground\n");
        printf("10.JBS Parade Ground\n");


}
void blue1()
```

```
{
        system("color 1");
        printf("\n11.Nagole\n12.Uppal\n13.Stadium\n14.NGRI\n15.Habsiguda\n16.Tarnaka\n
");
        printf("17.Mettuguda\n18.Seunderabad East\n19.Paradise\n20.Rasoolpura\n");
        printf("21.Prakash nagar\n22.Begumpet\n23.Ameerpet\n24.Madhura
Nagar\n25.Yusfguda\n");
        printf("26.RoadNO -5 Jubille hills\n27.Jubille hills check post\n28.Peddama gudi\n");
        printf("29.Madhapur\n30.Durgam Cheruvu\n31.Hitec City\n");
}
void red1()
{
        system("color 4");
        printf("\n32.LB Nagar\n33.Victoria
Memorial\n34.ChaitanyaPuri\n35.Dilsukhnagar\n");
        printf("36.Musarambagh\n37.New market\n38.Malakpet\n39.MG BUS\n40.Osmaina
Medical College\n");
        printf("41.Gandhi Bhavan\n42.Nampally\n43.Assembly\n44.Lakdhi-ka-pul\n");
        printf("45.Khairtabad\n46.Irrum manzil\n47.Panja Gutta\n48.Ameerpet\n");
        printf("49.S R Nagar\n50.Esi Hospital\n51.Erragadda\n52.Bharat
Nagar\n53.Moosapet\n");
        printf("54.Balanagar\n55.Kukatpally\n56.Kphb colony\n57.JNTU\n58.Miyapur\n");
}


void stations()
{
        int line;
        printf("Enter the Line you want to travel:");
        printf("\n1.Green\n");
        printf("2.Blue\n");
        printf("3.Red\n");
        scanf("%d",&line);
        switch(line)
        {
                case 1:{
                        system("cls");
                        green();
                        break;
                }
                case 2:{
                        system("cls");
                        blue();
                        break;
                }
                case 3:{
                        system("cls");
                        red();
                        break;
                }
                case 0:exit(0);
                default:printf("make correct choice:");


        }

}

void f(){
                newnode=(struct node *)malloc(sizeof(struct node));
                fflush(stdin);
                printf("\n enter name:");
                scanf("%s",newnode->data);
                fflush(stdin);
                printf("\n enter mobile number:");
                scanf("%s",newnode->number);
                newnode->link=NULL;
                if(start==NULL){
                        temp=newnode;
                        start=newnode;
                }
                else{
                        temp->link=newnode;
                        temp=newnode;
                }
}
void tra(){
        printf("\n----------------------------------------------------\n");
   printf("             TODAYS TRAVELLERS HISTORY                  \n");
   printf("\n----------------------------------------------------\n");

        if(start==NULL){
                printf("\nNo users had came.");
        }
        else{
                p=start;
                while(p!=NULL){
                        printf("\n%s - %s\t",p->data,p->number);
                        p=p->link;
                }
        }
        printf("\nEnter 1 to go to main\n0 to exit: ");
        int a;
```

```
	scanf("%d",&a);
	if(a==0)
		exit(0);
	else
		main();
}

void colors()
{

	printf("would you like to continue(1 or 0):");
	int num;
	scanf("%d",&num);
	if(num==1)
	{
	system("cls");

	main();
	}
	else if(num==0) {
		exit(0);
	}
	else{
		printf("invalid credentials");
	}
}

void buy()
{
	system("cls");
```

```
int dno,line;
printf("Enter the line you want to travel:\n1.green\n2.blue\n3.red\n");
scanf("%d",&line);
system("cls");
if(line==1)
{
green1();


}
else if(line==2)
{
        blue1();
}
else if(line==3)
{
        red1();
}
else
{
        printf("Invalid option:");
        buy();
}
printf("Enter source station number: ");
scanf("%d",&no);
system("cls");
system("color f");
printf("Enter Destination line you need to Travel: ");
printf(":\n1.green\n2.blue\n3.red\n");
scanf("%d",&line);
system("cls");
```

```
if(line==1)
{
green1();
}
else if(line==2)
{
        blue1();
}
else if(line==3)
{
        red1();
}
else
{
        printf("Invalid option:");
        buy();
}

printf("Enter Destination: ");
scanf("%d",&dno);
int dist;
dist=dno-no;
if(dist==0)
{
        buy();
}
else
{
        morebuy(dist);
}
```

```
}

void morebuy(int dist)
{
	system("cls");
	system("color f");
  int no;
	printf("Enter no of tickets you want to buy: ");
	scanf("%d",&no);

	if(no<=0)
	{
		printf("invalid number");
	}
	else
	{
		int cost=0;
		if(dist<0)
		{
			dist=dist*(-1);
		}
		if(dist<=5)
		{
			cost=no*10;
		}
		else if(dist<=10)
		{
			cost=no*20;
		}
```

```
            else if(dist<=15)
            {
                cost=no*30;
            }
            else if(dist<=20)
            {
                cost=no*40;
            }
            else if(dist>20)
            {
                cost=no*55;
            }
            while(no){
                f();
                no--;
            }

        ajaycash(cost);
            }


}

void ajaycash(int cost)
{
            system("cls");
            printf("Total cost = Rs.%d \n",cost);
            printf("\n Enter yor payement mode:\n");
            printf("1.cash\n2.Card\n");
            int ch;
```

```
                scanf("%d",&ch);

                if(ch==1)
                {
                 pay_cash();
                }
                else if(ch==2)
                {
                 pay_card(cost);
                }
                else
                {
                        printf("Invalid");
                        ajaycash(cost);
                }
}

void pay_card(int cost) {
        system("cls");
        char key[20];
        int flag=0, loc;
        card pc;
        FILE *f1,*f2;
        f2=fopen("temp.txt","w");
        f1=fopen("ajay.txt","r");
        printf("Enter your card number to pay\n");
        scanf("%s",key);

        char s,qwe[10],qw[20];
        int age,num1,bal;
```

```
s=getc(f1);
while(s!=EOF){
        if(s=='\n'){
                fscanf(f1,"%s",qwe);
                fscanf(f1,"%s",qw);
                fscanf(f1,"%d",&age);
                fscanf(f1,"%d",&num1);
                fscanf(f1,"%d",&bal);
        }
        if(strcmp(qwe,key)==0){
                flag=1;
                if(bal<cost) {
                        printf("Insufficient balance! Please pay via cash\n");
                        pay_cash();
                        break;
                }
                printf("Remaining Balance: %d\n",bal);
                bal-=cost;
                printf("Current Balance: %d\n",bal);
                printf("Payment Successful! Enjoy your trip\n");
                flag=1;
                fprintf(f2,"\n%s\t%s\t%d\t%d\t%d",qwe,qw,age,num1,bal);
                break;
        }
        fprintf(f2,"\n%s\t%s\t%d\t%d\t%d",qwe,qw,age,num1,bal);
        s=getc(f1);
}
remove("ajay.txt");
rename("temp.txt","ajay.txt");
if(flag==0){
		printf("Card not found!\n");
		ajaycash(cost);
	}
	fclose(f1);
	fclose(f2);
	time1();
	printf("\nwould you like to continue(1 or 0):");
	int num;
	scanf("%d",&num);
	if(num==1)
	{
		system("cls");

		main();
	}
	else
		exit(0);
}
void pay_cash() {


	int tokno;
	printf("Please collect and submit the printed token number at the counter and pay your
bill\n");
	tokno=rand();
	printf("Your token number:%d",tokno);
	time1();
	printf("\nwould you like to continue:1 or 0");
	int num;
	scanf("%d",&num);
	if(num==1)
	{
```

```
                system("cls");

                main();
        }
        else
                exit(0);
}
void create_card() {
        system("cls");
        FILE *f;
        card c;
        char email[20];
        f=fopen("ajay.txt","a");
        printf("Enter your name\n");
        scanf("%s",c.name);
        printf("Enter your age\n");
        scanf("%d",&c.age);
                while(c.age<3){
                printf("Incorrect age! Please enter again\n");
                scanf("%d",&c.age);
        }
        printf("Enter your Phone number\n");
        scanf("%s",c.number);
        c.num[0]=c.name[0];
        c.num[1]=c.name[1];
        c.num[2]=c.number[0];
        c.num[3]=c.number[1];
        fprintf(f,"\n%s\t",c.num);
        printf("Enter the amount you want to recharge\n");
        scanf("%d",&c.balance);
```

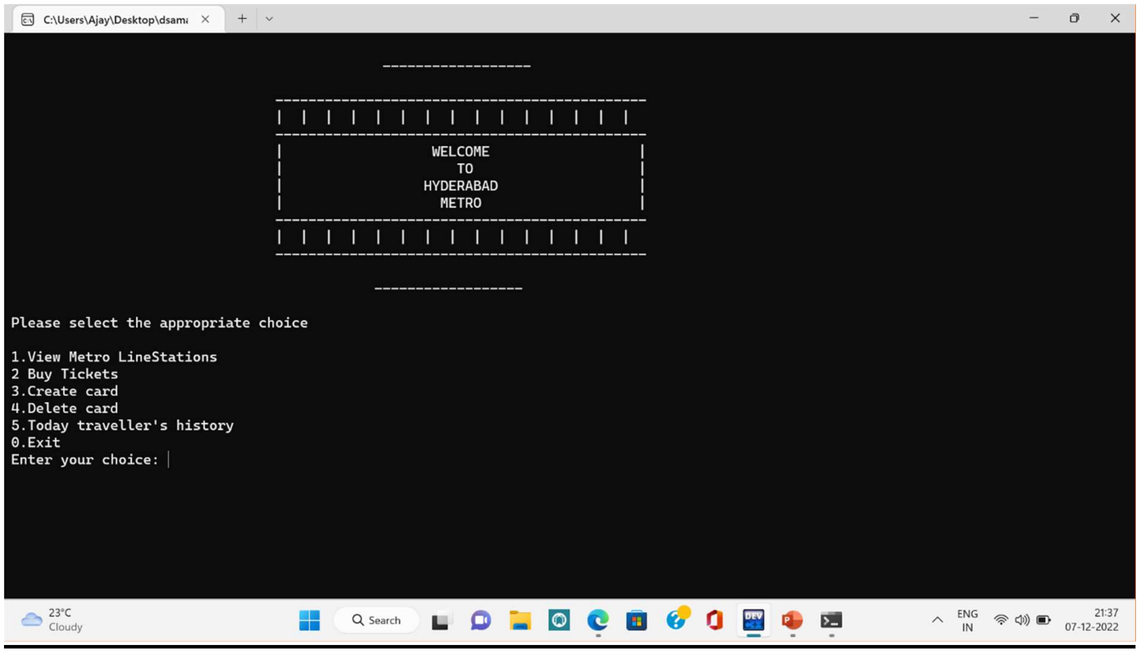
```
	printf("Enter your email-ID\n");
	scanf("%s",email);
	printf("CARD DETAILS\n");
	printf("Name:  %s\n",c.name);
	printf("Age:     %d\n",c.age);
	printf("Phone No.:       %s\n",c.number);
	printf("Card number:  %s\n",c.num);
	printf("Balance:          %d\n",c.balance);
	if(c.age>=60)
		printf("Type:   Senior Card\n");
	else
		printf("Type:   Normal Card\n");
	printf("Card details successfully sent to your phone number and email");
	fprintf(f,"%s\t%d\t%s\t%d",c.name,c.age,c.number,c.balance);
	fclose(f);
	printf("\nwould you like to contin 1 or 0");
	int num;
	scanf("%d",&num);
	if(num==1)
	{
		system("cls");
		main();
	}
	else
	{
		exit(0);
	}
	getchar();
}
```

```
void delete_card() {
        system("cls");
        char key[20];
        int flag=0;
        char cd[5];
        card d;
        FILE *f1,*f2;
        f1=fopen("ajay.txt","r");
        f2=fopen("temp.txt","w");
        printf("Cancellation charges = Rs.5\n");
        printf("Enter the card number to be deleted\n");
        scanf("%s",key);
        char s,qwe[10],qw[20];
        int age,num1,bal,bal1;

        s=getc(f1);
        while(s!=EOF){
                if(s=='\n'){
                        fscanf(f1,"%s",qwe);
                        fscanf(f1,"%s",qw);
                        fscanf(f1,"%d",&age);
                        fscanf(f1,"%d",&num1);
                        fscanf(f1,"%d",&bal);
                }
                if(strcmp(qwe,key)==0){
                        flag=1;
                        bal1=bal;
                }
                else
                        fprintf(f2,"\n%s\t%s\t%d\t%d\t%d",qwe,qw,age,num1,bal);
```

```
            s=getc(f1);
        }
        fclose(f1);
        fclose(f2);
        remove("ajay.txt");
        rename("temp.txt","ajay.txt");
        if(flag==0)
            printf("Card not found!\n");
        else {
            printf("Remaining Balance in the card = Rs.%d\n",bal1);
            bal1-=100;
            printf("Card deleted successfully! Please collect your cash of Rs.%d\n",bal1);
        }
        printf("\nwould you like to contin 1 or 0");
        int num;
        scanf("%d",&num);
        if(num==1)
        {
            system("cls");
            main();
        }
        else
        {
            exit(0);
        }

}
void time1()
{
        time_t timeNow;
        struct tm* time_info;
    char timeStr[sizeof"HH:MM"];
        time(&timeNow);
        timeNow += 60*(5);
    time_info = localtime(&timeNow);
    strftime(timeStr, sizeof(timeStr), "%H:%M", time_info);
    printf("\nThe next train will arrive in : %s\n", timeStr);
}




main()
{   system("cls");
    system("color f");
    printf("\n\t\t\t                  ------------------\n");
    printf("\n\t\t\t       ---------------------------------------------");
    printf("\n\t\t\t       | | | | | | | | | | | | | | | |  ");
    printf("\n\t\t\t       ---------------------------------------------");
    printf("\n\t\t\t       |              WELCOME              |");
    printf("\n\t\t\t       |                TO              |");
    printf("\n\t\t\t       |             HYDERABAD              |");
    printf("\n\t\t\t       |              METRO              |");
    printf("\n\t\t\t       ---------------------------------------------");
    printf("\n\t\t\t       | | | | | | | | | | | | | | | |  ");
    printf("\n\t\t\t       ---------------------------------------------");
    printf("\n");
    printf("\n\t\t\t                 ------------------\n");
        printf("\nPlease select the appropriate choice\n");
        printf("\n1.View Metro LineStations\n");
        printf("2 Buy Tickets\n");
        printf("3.Create card\n");
        printf("4.Delete card\n");
        printf("5.Today traveller's history\n");
        printf("0.Exit\n");
        int ch;
        printf("Enter your choice: ");
        scanf("%d",&ch);
        switch(ch)
        {
                case 1:{
                        stations();
                        break;
                }
                case 2:{
                        buy();
                        break;
                }
                case 3:{
                        create_card();
                        break;
                }
                case 4:{
                        delete_card();
                        break;
                }
                case 5:{
```

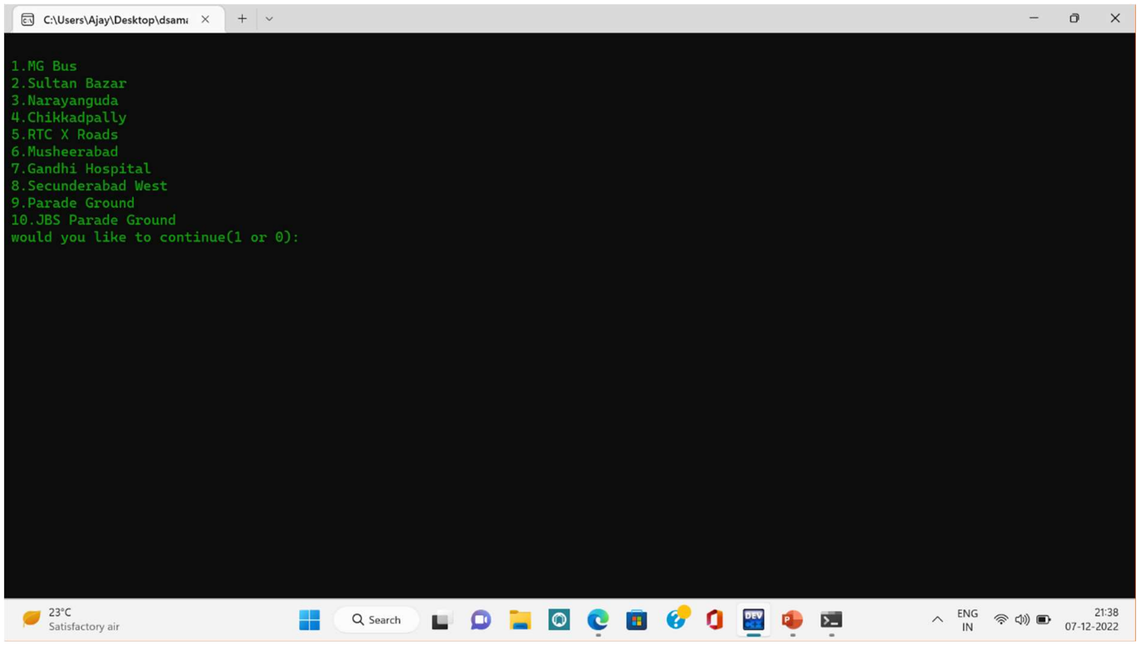
```
                        tra();
                        break;
                }
                case 0:exit(0);
                                break;
                default:printf("Invalid choice\n");

        }
}
```

OUTPUT:

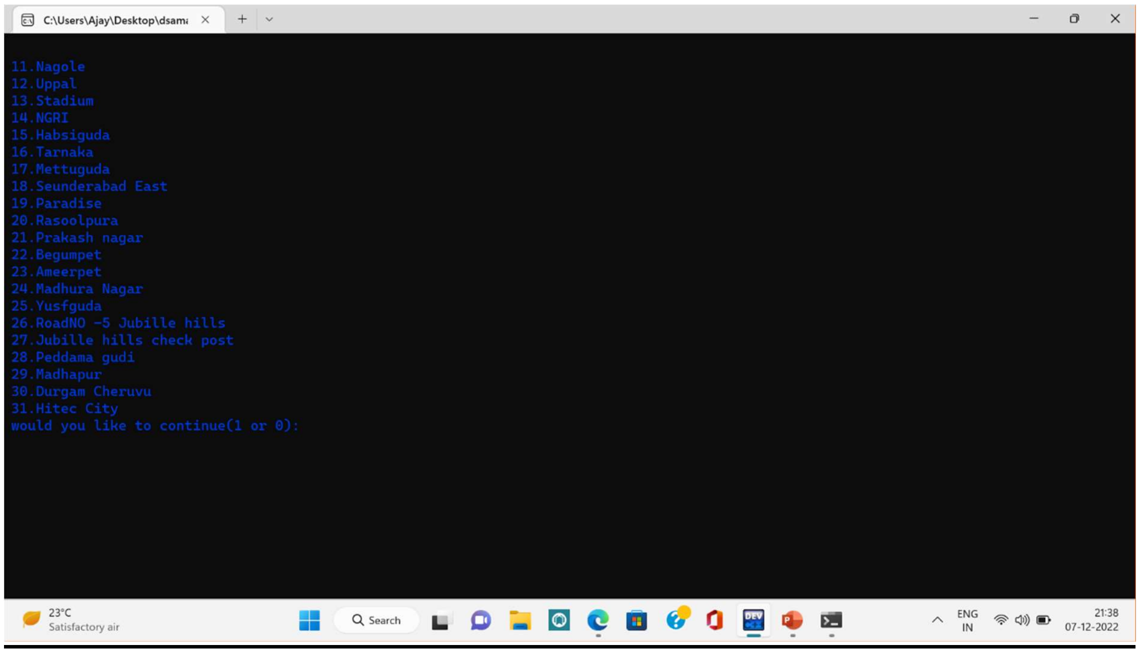


```
                                  ------------------

                     ---------------------------------------------
                     | | | | | | | | | | | | | | |
                     ---------------------------------------------
                     |                WELCOME                    |
                     |                   TO                      |
                     |               HYDERABAD                   |
                     |                 METRO                     |
                     ---------------------------------------------
                     | | | | | | | | | | | | | | |
                     ---------------------------------------------

                                 ------------------

Please select the appropriate choice

1.View Metro LineStations
2 Buy Tickets
3.Create card
4.Delete card
5.Today traveller's history
0.Exit
Enter your choice:
```

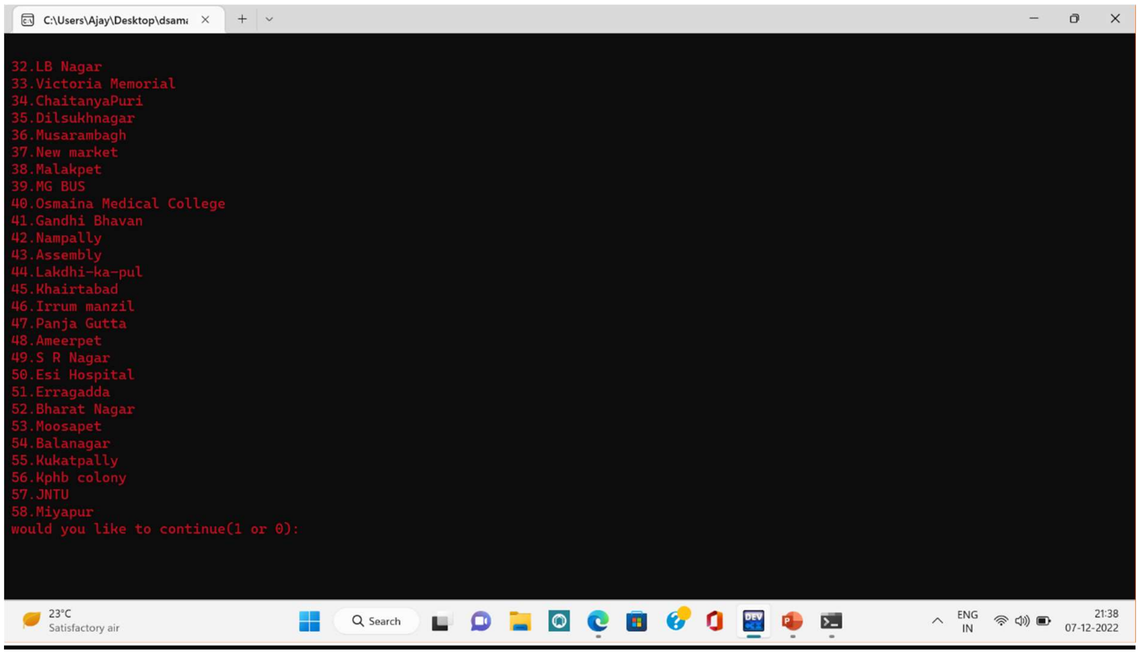
VIEW STATIONS AND LINES



```
1.MG Bus
2.Sultan Bazar
3.Narayanguda
4.Chikkadpally
5.RTC X Roads
6.Musheerabad
7.Gandhi Hospital
8.Secunderabad West
9.Parade Ground
10.JBS Parade Ground
would you like to continue(1 or 0):
```

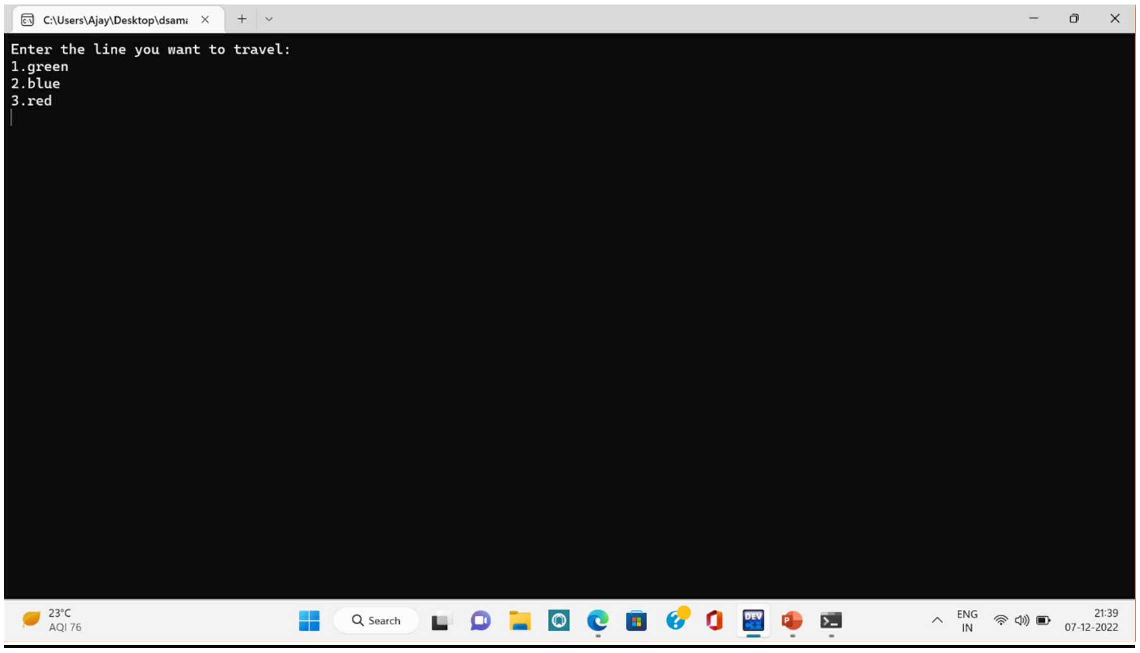


```
11.Nagole
12.Uppal
13.Stadium
14.NGRI
15.Habsiguda
16.Tarnaka
17.Mettuguda
18.Seunderabad East
19.Paradise
20.Rasoolpura
21.Prakash nagar
22.Begumpet
23.Ameerpet
24.Madhura Nagar
25.Yusfguda
26.RoadNO -5 Jubille hills
27.Jubille hills check post
28.Peddama gudi
29.Madhapur
30.Durgam Cheruvu
31.Hitec City
would you like to continue(1 or 0):
```



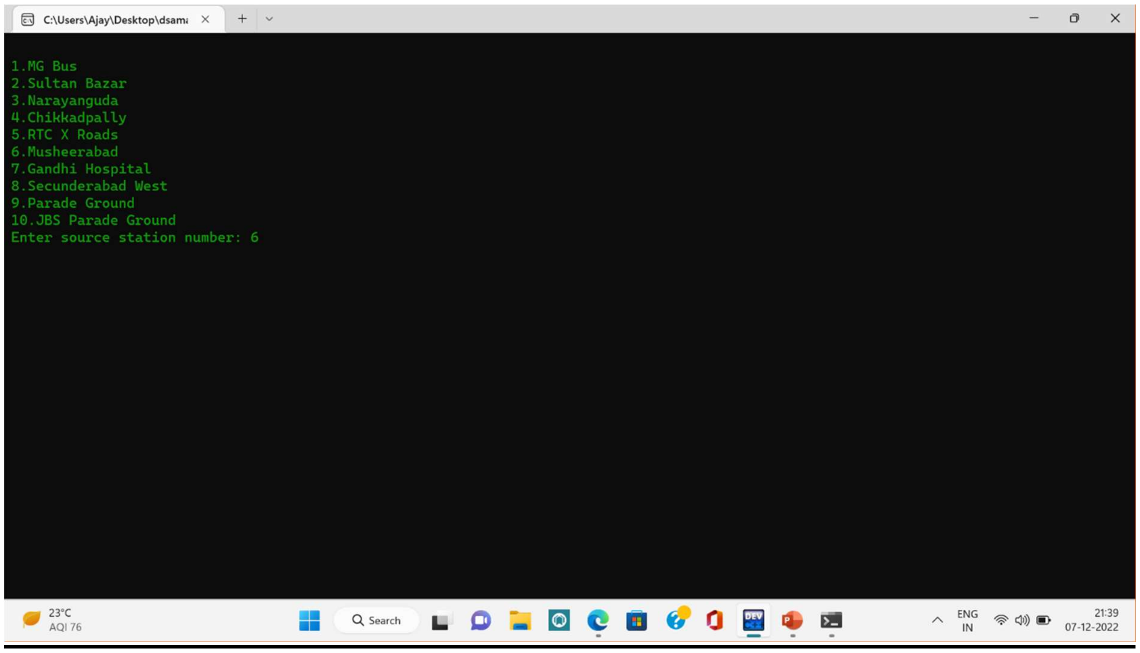
```
32.LB Nagar
33.Victoria Memorial
34.ChaitanyaPuri
35.Dilsukhnagar
36.Musarambagh
37.New market
38.Malakpet
39.MG BUS
40.Osmaina Medical College
41.Gandhi Bhavan
42.Nampally
43.Assembly
44.Lakdhi-ka-pul
45.Khairtabad
46.Irrum manzil
47.Panja Gutta
48.Ameerpet
49.S R Nagar
50.Esi Hospital
51.Erragadda
52.Bharat Nagar
53.Moosapet
54.Balanagar
55.Kukatpally
56.Kphb colony
57.JNTU
58.Miyapur
would you like to continue(1 or 0):
```

BUY TICKETS



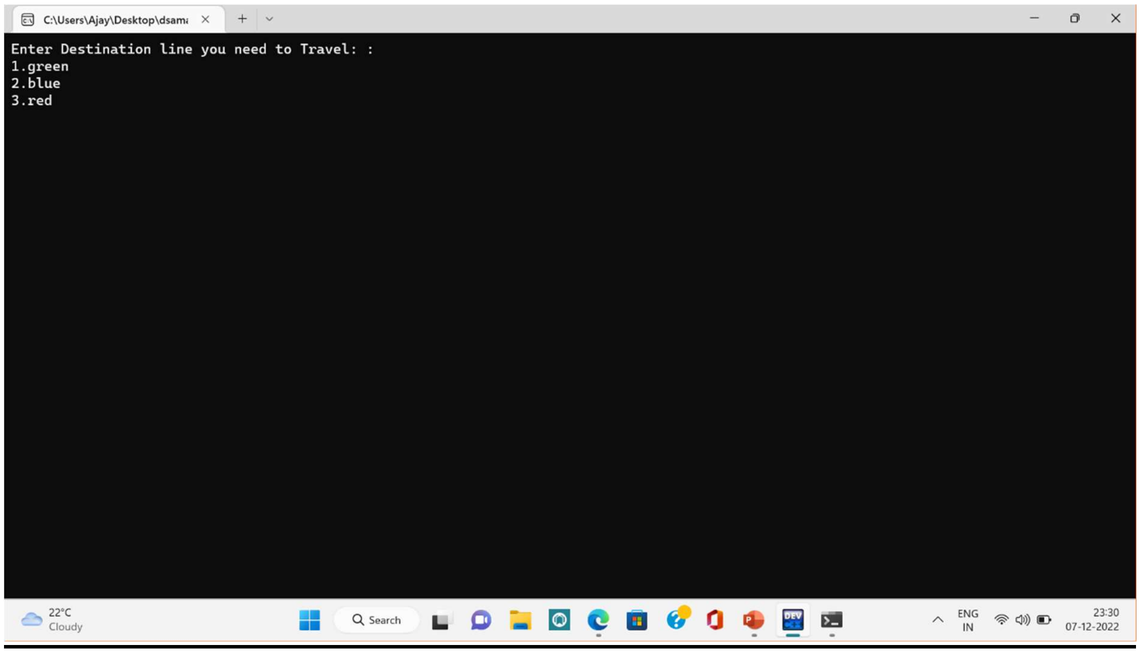
```
Enter the line you want to travel:
1.green
2.blue
3.red
```

SELECTING SOURCE LINE

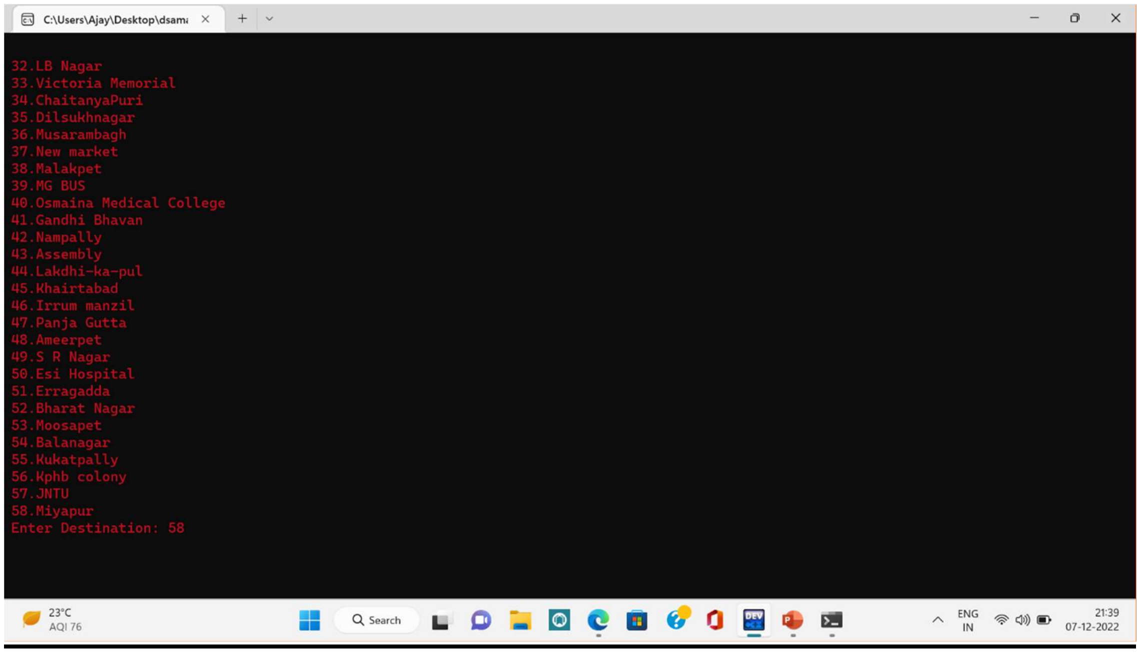


```
1.MG Bus
2.Sultan Bazar
3.Narayanguda
4.Chikkadpally
5.RTC X Roads
6.Musheerabad
7.Gandhi Hospital
8.Secunderabad West
9.Parade Ground
10.JBS Parade Ground
Enter source station number: 6
```

SELECTING DESTINATION

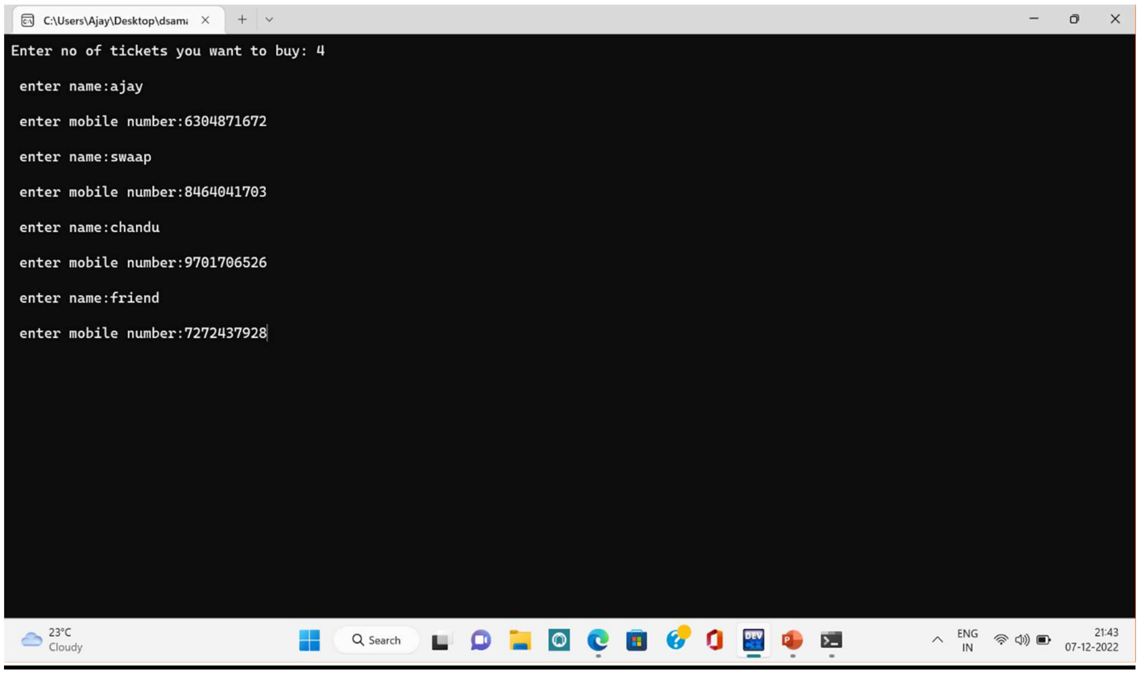


```
Enter Destination line you need to Travel: :
1.green
2.blue
3.red
```



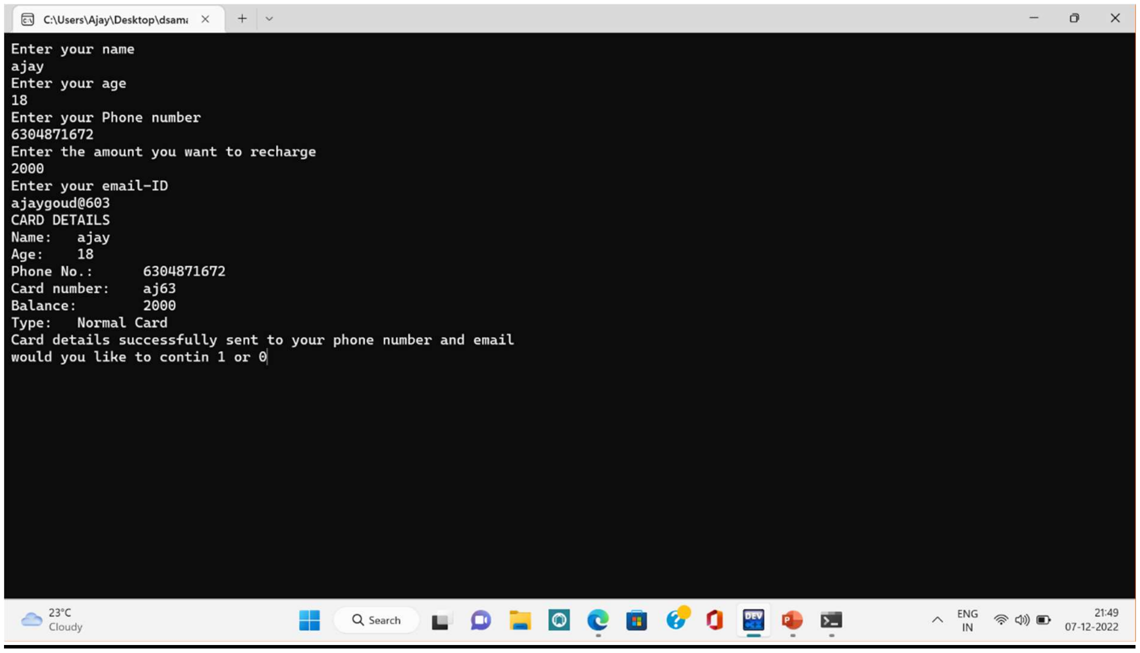
```
32.LB Nagar
33.Victoria Memorial
34.ChaitanyaPuri
35.Dilsukhnagar
36.Musarambagh
37.New market
38.Malakpet
39.MG BUS
40.Osmaina Medical College
41.Gandhi Bhavan
42.Nampally
43.Assembly
44.Lakdhi-ka-pul
45.Khairtabad
46.Irrum manzil
47.Panja Gutta
48.Ameerpet
49.S R Nagar
50.Esi Hospital
51.Erragadda
52.Bharat Nagar
53.Moosapet
54.Balanagar
55.Kukatpally
56.Kphb colony
57.JNTU
58.Miyapur
Enter Destination: 58
```

NUMBER OF TICKETS AND THEIR INFORMATION



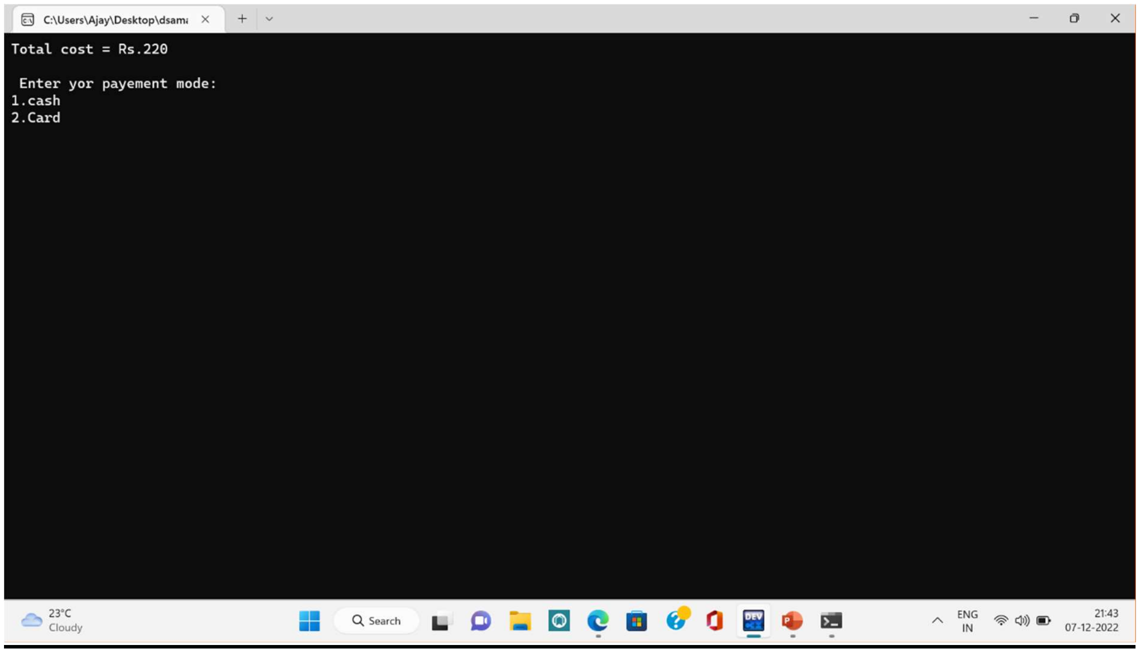
```
Enter no of tickets you want to buy: 4

 enter name:ajay

 enter mobile number:6304871672

 enter name:swaap

 enter mobile number:8464041703

 enter name:chandu

 enter mobile number:9701706526

 enter name:friend

 enter mobile number:7272437928
```

CREATING OF NEW CARD

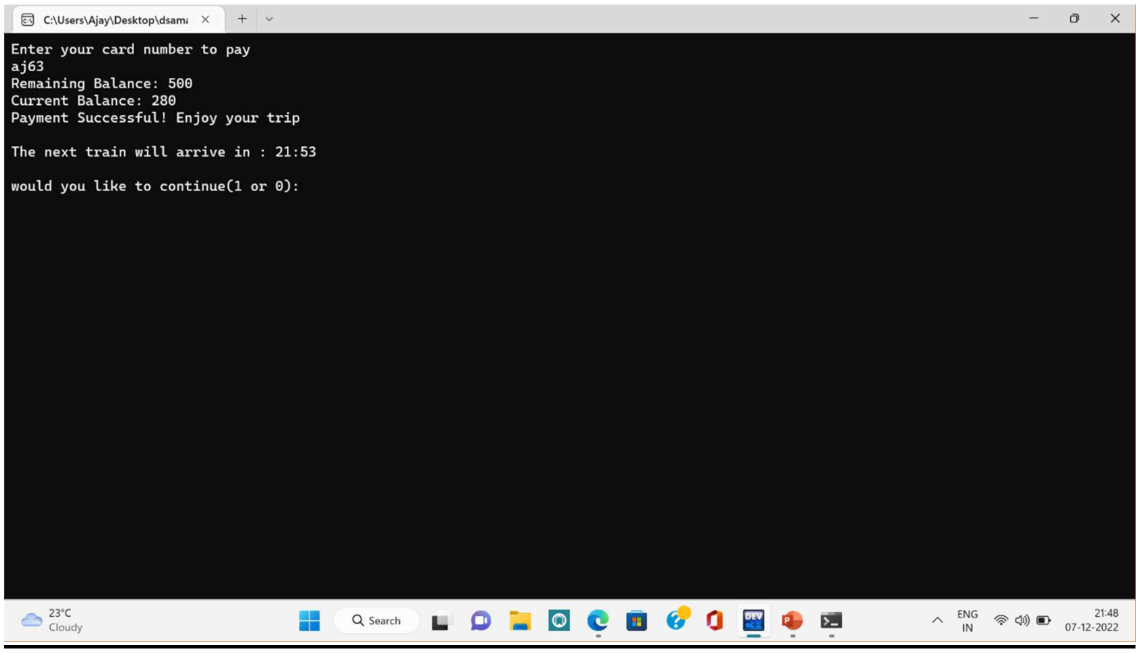


```
Enter your name
ajay
Enter your age
18
Enter your Phone number
6304871672
Enter the amount you want to recharge
2000
Enter your email-ID
ajaygoud@603
CARD DETAILS
Name:   ajay
Age:    18
Phone No.:      6304871672
Card number:    aj63
Balance:        2000
Type:   Normal Card
Card details successfully sent to your phone number and email
would you like to contin 1 or 0
```

PAYMENT MODE

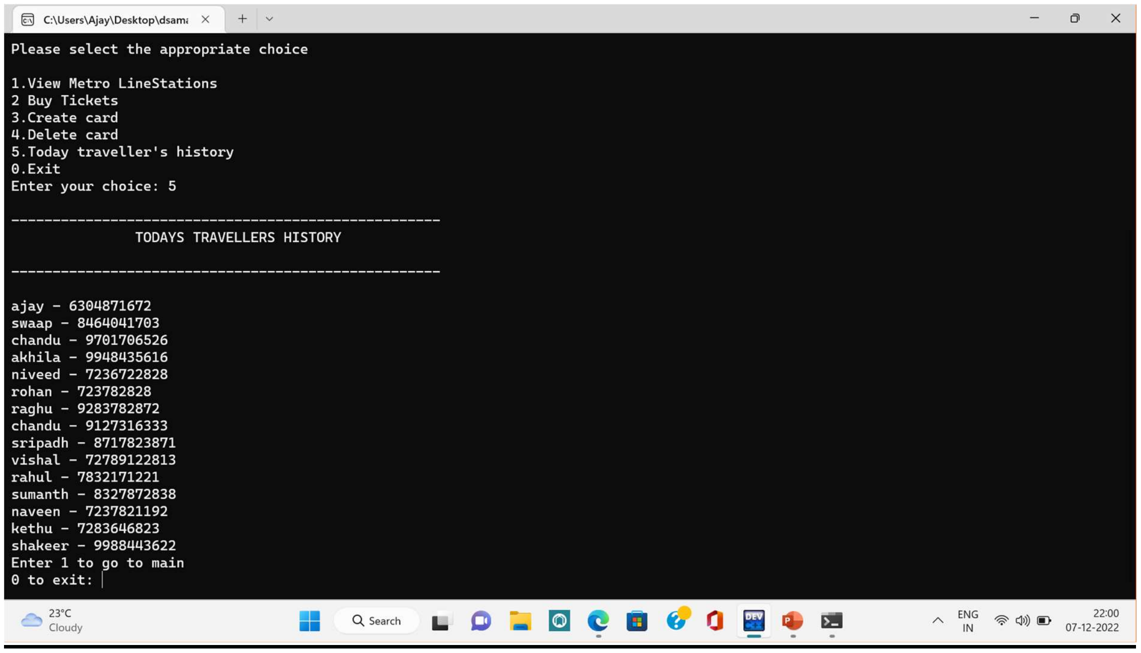


```
Total cost = Rs.220

 Enter yor payement mode:
1.cash
2.Card
```



```
Enter your card number to pay
aj63
Remaining Balance: 500
Current Balance: 280
Payment Successful! Enjoy your trip

The next train will arrive in : 21:53

would you like to continue(1 or 0):
```

TRAVELERS HISTORY



```
Please select the appropriate choice

1.View Metro LineStations
2 Buy Tickets
3.Create card
4.Delete card
5.Today traveller's history
0.Exit
Enter your choice: 5

----------------------------------------------------
               TODAYS TRAVELLERS HISTORY

----------------------------------------------------

ajay - 6304871672
swaap - 8464041703
chandu - 9701706526
akhila - 9948435616
niveed - 7236722828
rohan - 723782828
raghu - 9283782872
chandu - 9127316333
sripadh - 8717823871
vishal - 72789122813
rahul - 7832171221
sumanth - 8327872838
naveen - 7237821192
kethu - 7283646823
shakeer - 9988443622
Enter 1 to go to main
0 to exit:
```

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.

- In future changes can be done by linking the admin's mobile number for more security.

References

C Language:-

C Programming Language – GeeksforGeeks

Data Structures:-

Data Structures - GeeksforGeeks