

GAS BILLING APPLICATION

A Course Project report submitted
in partial fulfillment of requirement for the award of degree

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

By

M. Pranavi	(2203A52L02)
K. Vasanth	(2203A52L03)
K. Achyuthreddy	(2203A52L04)
N. Sanjay	(2203A52L05)

Under the guidance of

Mr. SRINIVAS O

Assistant Professor, Department of CSE.



Department of Computer Science and Artificial Intelligence



Department of Computer Science and Artificial Intelligence

CERTIFICATE

This is to certify that project entitled “**GAS BILLING APPLICATION**” is the bonafied work carried out by **M. Pranavi, K, Vasanth, K. Achyuthreddy, N. Sanjay** as a Course Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING** during the academic year 2022-2023 under our guidance and Supervision.

Mr. SRINIVAS O

Asst. Professor,
S R University,
Ananthasagar, Warangal.

Dr. M. Sheshikala

Assoc. Prof. & HOD (CSE),
S R University,
Ananthasagar, Warangal.

ACKNOWLEDGEMENT

We express our thanks to Course co-coordinator **Mr. SRINIVAS O, Asst. Prof.** for guiding us from the beginning through the end of the Course Project. We express our gratitude to Head of the department CS&AI, **Dr. M. Sheshikala, Associate Professor** for encouragement, support and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, School of Computer Science and Artificial Intelligence, **Dr C. V. Guru Rao**, for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

ABSTRACT

This project is a Gas connection system written in the C language that uses a double linked list. The system allows users to new connection enrollment, terminate current connection, book a cylinder, employee, feedback, and exit. It also includes features such as alphabetically and displaying the number of contacts stored in the directory. All data is stored in a linked list, which makes the system fast and efficient. The user interface is designed to be straightforward and intuitive. The code is modular, well-documented, and designed to be easily extensible. This project is an ideal starting point for any application that requires a gas booking system. This project is an implementation of a gas booking system using the C programming language and a double linked list data structure. It provides the user with an efficient way to store, search, and modify contact information. The program has functions to add, delete, and modify contacts, as well as to search for contacts by name or phone number. It also has a utility to display all contacts in the connection enrollment. The double linked list data structure enables efficient searching and modification of contact information, and booking, delete current connection while providing a compact memory footprint. The program is written in C and is well documented, making it easy to modify and extend.

Table of Contents
GAS BILLING APPLICATION

I. Introduction	1 - 3
1.1 Problem Statement	1
1.2 System Design & Modules	2
1.3 Module Details	2-3
1.4 Software Specifications	3
II. Coding	3-17
III. Outputs	18-20
IV. Conclusion	21

1.INTRODUCTION

This project is a Gas connection system written in the C language that uses a double linked list. The system allows users to new connection enrollment, terminate current connection, book a cylinder, employee, feedback, and exit. It also includes features such as alphabetically and displaying the number of contacts stored in the directory. All data is stored in a linked list, which makes the system fast and efficient. The user interface is designed to be straightforward and intuitive. The code is modular, well-documented, and designed to be easily extensible. This project is an ideal starting point for any application that requires a gas booking system. This project is an implementation of a gas booking system using the C programming language and a double linked list data structure. It provides the user with an efficient way to store, search, and modify contact information. The program has functions to add, delete, and modify contacts, as well as to search for contacts by name or phone number. It also has a utility to display all contacts in the connection enrollment. The double linked list data structure enables efficient searching and modification of contact information, and booking, delete current connection while providing a compact memory footprint. The program is written in C and is well documented, making it easy to modify and extend.

1.1) PROBLEM STATEMENT

Design a C language program using a single linked list to create and book a cylinder. A gas booking system should give new connection enrollment. The program should allow a user to new connection enrollment, terminate current connection, book a cylinder, feedback. The program should also be able to print out the entire gas booking system.

1.2 System Design & Modules

The Gas booking System will be designed using C language with a single linked list. It will provide the facility of getting information number of cylinders and total price of cylinders. And it will store contact information such as names, phone numbers, and address. This will enable users to search for a contact's information quickly and efficiently.

1.3 Modules:

New connection enrollment: This module will allow users to add the names of users to the connection process. It will take in the contact's name, phone number and address, this module is created upon the basis of create a node in **double linked list**.

Terminate current connection: This module will allow users to Delete a contact by service number. It will Terminate the contact's name and address. This module is created upon the basis of **delete a node in double linked list**.

Book a cylinder: This module will allow users to make booking number of cylinders and total price of cylinders booked and easier for gas booking to the customers. This module is created upon the basis of **inserting data into double linked list, printing the data**.

Employee: This module is to check employee feedbacks and connections.

Feedback: This module will allow users to give feedback is to improve the situation or the person's performance. This module is created upon the basis of **Insert data at a specific node**.

Module Details

1)New connection enrollment:

This module allows users to add a details like name, mobile number, house number, street, village, town.

2)Terminate current connection:

This module will allow users to delete the details from the connection enrollment by entering service number to the given customer.

3)Book a cylinder:

This project GAS BOOKING SYSTEM includes some facilities for the customer to get the required number of cylinders. And it will provide the facility of getting information number of subsidy cylinders left. This software keeps the records of all

the customers and their information about how many time the booking of cylinders done.

Objective of this software is to computerize the manual system of a gas agency, so that all the transactions become fast and easy. It replaces all the paper work. It works according to the need and lowers the workload of user.

Objective of the program rather sample.it provides easy management of transaction that are regular in business concerned for which it has been developed. It's providing an easy navigation menu which a layman user can also use.

It manages the entire record along whit date, item code, item cost etc.in this program there is a working mode available administrator.an administrator is as usual given all the rights like adding information, modifying existing information, deleting and viewing the information.

In this system clearly aims at reducing the paper work at the cost to be borne of the registers and the other things are much higher than the system's working. Its keep records of all the information given to it according to the date which makes locating a record quite easy.

4)Feedback: This module allows users to helpful information or criticism that is given to someone to say what can be done to improve a person's performance of a task.

1.4 System requirements

System Requirements:

1. Operating System: The system requires a computer with a modern operating system such as Windows, Mac OS, or Linux.
2. Programming Language: The system should be written in C language.
3. Data Structures: The system should use Double linked list to store the data.
4. User Interface: The system should provide a user-friendly interface to allow the user to view, add, delete, and modify entries in the to fill the details.
5. Memory: The system should be able to store at least 5000 entries in the directory.
6. Input: The system should accept user input from the keyboard.
7. Output: The system should be able to display the Gas booking system in a readable format.
8. Documentation: The system should be properly documented.

CODE:

```
#include<stdio.h>
#include<stdlib.h>
struct Node{
    char name[100];
    char hmno[100];
    char street[100];
    char village[100];
    char town[100];
    char feedback[1000];
    int fb;
    long serno;
    long mno;
    int cylinder;
    struct Node *prev;
    struct Node *next;
};
struct Node *start;
struct Node *temp,*newnode,*ptr;
int i,h;
    long a=0;
void bill();
void newenrol();
void terminate();
void feedback();
void viewfeedback();
void create();
void connections();
void employee();
void main()
{
    int id,ch;
    system("cls");
```

```

printf("\n 1.NEW CONNECTION ENROLLMENT");
printf("\n 2.TERMINATE CURRENT CONNECTION");
printf("\n 3.BOOK A CYLINDER");
printf("\n 4.FEEDBACK");
printf("\n 5.EMPLOYEE");
printf("\n 6.EXIT");
while(1){
    printf("\n Make your choice:");
    scanf("%d",&ch);
    switch(ch)
    {
        case 1:
            newenrol();
            break;
        case 2:
            terminate();
            break;
        case 3:
            bill();
            break;
        case 4:
            feedback();
            break;
        case 5:
            printf("\nPLEASE ENTER YOUR EMPLOYEE ID:");
            scanf("%d",&id);
            if(id==1||id==2){
                employee();
            }
            else if(id==3||id==4){
                employee();
            }
            else{
                printf("\nERROR! INVALID EMPLOYEE ID");
            }
        }
    }
}

```

```

        }
        break;
    case 6:
        exit(0);
        break;
    default:
        printf("\n Invalid choice");
        break;
    }
}
}
void create()
{
    int total,index,item,id;
    if(start==NULL)
    {
        printf("\n Enter no.of connections:");
        scanf("%d",&total);
        if(total<=0)
        {
            printf("\n Total no.of connections need to be Positive");
        }
        else
        {
            temp = (struct Node*)malloc(sizeof(struct Node));
            printf("\n Enter the person details:");
            printf("\n NAME:");
            scanf("%s",temp->name);
            printf("\n MOBILE NUMBER:");
            scanf("%ld",&temp->mno);
            printf("\n HOUSE NO:");
            scanf("%s",temp->hmno);
            printf("\n STREET:");
            scanf("%s",temp->street);

```

```

printf("\n VILLAGE:");
scanf("%s",temp->village);
printf("\n TOWN:");
scanf("%s",temp->town);
temp->fb=0;
temp->cylinder=0;
temp->serno=a;
temp->prev=NULL;
temp->next=NULL;
start = temp;
temp->serno=temp->serno+1;
printf("\nCONNECTION    SUCCESSFULLY    REGISTERED
WITH-");

printf("\n SERVICE NO (SERNO):%ld",temp->serno);
a++;
for(index=2;index<=total;index++)
{
    newnode = (struct Node*)malloc(sizeof(struct Node));
    newnode->serno=a;
    printf("\n Enter the person details:");
    printf("\n NAME:");
    scanf("%s",newnode->name);
    printf("\n MOBILE NUMBER:");
    scanf("%ld",&newnode->mno);
    printf("\n HOUSE NO:");
    scanf("%s",newnode->hmno);
    printf("\n STREET:");
    scanf("%s",newnode->street);
    printf("\n VILLAGE:");
    scanf("%s",newnode->village);
    printf("\n TOWN:");
    scanf("%s",newnode->town);
    newnode->fb=0;
    newnode->cylinder=0;

```

```

        newnode->next = NULL;
        newnode->prev=temp;
        temp->next = newnode;
        temp=newnode;
        newnode->serno=newnode->serno+1;
        printf("\nCONNECTION    SUCCESSFULLY    REGISTERED
WITH-");

        printf("\n SERVICE NO (SERNO):%ld",newnode->serno);
        a++;
    }
}
else
{
    printf("\n CONNECTIONS ALREADY REGISTERED");
}
}
void bill()
{
    struct Node *temp;
    long us;
    float bill;
    int n,h,k;
    if(start==NULL)
    {
        printf("\n error! no enrolled connections");
    }
    else
    {
        printf("ENTER YOUR SERVICE NUMBER:");
        scanf("%ld",&us);
        temp=start;
        while(temp!=NULL)
        {

```

```

        if(us==temp->serno){
            printf("\n    ENTER    THE    NO.OF    CYLINDERS
REQUIRED");

            scanf("%d",&n);
            printf("\n ENTER 1 TO CONFIRM BOOKING ");
            scanf("%d",&k);
            if(k==1){
                temp->cylinder=temp->cylinder+n;
                bill=n*1124;
                printf("\n BOOKING SUCCESSFULL");
                for(i=0;i<7;i++){
                    printf(".\t");
                    sleep(1);
                }
                system("cls");
                printf("                BHARAT GAS

");

                printf("\n-----");
                printf("\nSERVICE NO    :%ld",temp->serno);
                printf("\nNAME    :%s",temp->name);
                printf("\nH.NO    :%s",temp->hmno);
                printf("\nADDRESS :%s,",temp->street);
                printf("\n    %s,",temp->village);
                printf("\n    %s",temp->town);
                printf("\nMOBILE NO :%ld",temp->mno);
                printf("\n-----");
                printf("\nNO.OF CYLINDERS    :%d",temp->cylinder);
                printf("\nBASE RATE    :%d",720*n);
                printf("\nCENTRAL GST    :%d",18*n);
                printf("\nSTATE GST    :%d",18*n);
                printf("\nTOTAL BILL    :%d",bill);
                printf("\n-----");
            }
            temp=temp->next;

```

```

    }
    printf("\n PRESS 1 TO GO BACK");
    scanf("%d",&h);
    if(h==1){
        main();
    }
}
}

void newenrol()
{
    newnode=(struct Node *)malloc(sizeof(struct Node));
    newnode->serno=a;
    printf("\n PLEASE FILL THE FOLLOWING DETAILS FOR NEW
CONNECTION:");
    printf("\n NAME:");
    scanf("\n");
    scanf("%[^\\n]*c",newnode->name);
    printf(" MOBILE NUMBER:");
    scanf("%ld",&newnode->mno);
    printf(" HOUSE NO:");
    scanf("%s",newnode->hmno);
    printf(" STREET:");
    scanf("\n");
    scanf("%[^\\n]*c",newnode->street);
    printf(" VILLAGE:");
    scanf("\n");
    scanf("%[^\\n]*c",newnode->village);
    printf(" TOWN:");
    scanf("\n");
    scanf("%[^\\n]*c",newnode->town);
    newnode->fb=0;
    newnode->cylinder=0;
    newnode->next=NULL;

```

```

        newnode->prev=NULL;
    if(start==NULL){
        start=newnode;
        newnode->serno=newnode->serno+1;
        printf("\nCONNECTION    SUCCESSFULLY    REGISTERED
WITH-");
        printf("\nSERVICE NO (SERNO)    :%ld",newnode->serno);
        a++;
    }
    else
    {
        temp=start;
        while(temp->next!=NULL)
        {
            temp=temp->next;
        }
        temp->next=newnode;
        newnode->prev=temp;
        newnode->serno=newnode->serno+1;
        printf("\nCONNECTION    SUCCESSFULLY    REGISTERED
WITH-");
        printf("\nSERVICE NO (SERNO)    :%ld",newnode->serno);
        a++;
    }
}
void terminate()
{
    int loc=1,count=1,i=0,h,a;
    long us;
    system("cls");
    if(start==NULL)
    {
        printf("\n error! no enrolled connections");
    }
}

```



```

else
{
    printf("ENTER YOUR SERVICE NUMBER:");
    scanf("%ld",&us);
    temp=start;
    while(temp!=NULL)
    {
        i++;
        temp=temp->next;
    }
    temp=start;
while(temp!=NULL)
    {
        if(us==temp->serno){
            break;
        }
        loc++;
        temp=temp->next;
    }
    if(loc==1){
        if(start->next==NULL)
        {
            printf("\n1.CONIFIRM DELETE\n2.SKIP\n");
            scanf("%d",&a);
            if(a==1){
                for(i=0;i<7;i++){
                    printf(".\t");
                    sleep(1);
                }
                printf("\n THE CONECTION WITH SERVICE NO %ld IS
DELETED",start->serno);
                start=NULL;
            }
        }
    }
}

```

```

else{
    printf("\n1.CONFIRM DELETE\n2.SKIP\n");
    scanf("%d",&a);
    if(a==1){
        for(i=0;i<7;i++){
            printf(".\t");
            sleep(1);
        }

        temp=start;
        printf("\n THE CONECTION WITH SERVICE NO %ld IS
DELETED",start->serno);
        start = temp->next;
        start = start->next;
        start->prev=NULL;
        free(temp);
    }
}
else if(loc==i&&loc!=1){
    printf("\n1.CONFIRM DELETE\n2.SKIP\n");
    scanf("%d",&a);
    if(a==1){
        for(i=0;i<7;i++){
            printf(".\t");
            sleep(1);
        }

        temp=start;
        while(temp->next!=NULL)
        {
            ptr=temp;
            temp=temp->next;
        }
        ptr->next=NULL;
        printf("\n THE CONECTION WITH SERVICE NO %ld IS

```

```

DELETED",temp->serno);
    free(temp);
}
}
else{
    temp=start;
    while(temp!=NULL&&count!=loc)
    {
        ptr=temp;
        temp=temp->next;
        count++;
    }
    if(temp==NULL)
    {
        printf("\n NO CONNECTION IS PRESENT");
    }
    else
    {
        printf("\n1.CONFIRM DELETE\n2.SKIP\n");
        scanf("%d",&a);
        if(a==1){
            for(i=0;i<7;i++){
                printf(".\t");
                sleep(1);
            }
            ptr->next=temp->next;
            printf("\n THE CONECTION WITH SERVICE NO %d IS
DELETED",temp->serno);
            free(temp);
        }
    }
}
printf("\n PRESS 1 TO GO BACK");

```

```

        scanf("%d",&h);
        if(h==1){
            main();
        }
    }
void feedback()
{
    long us;
    system("cls");
    printf("\n ENTER YOUR SERVICE NO:");
    scanf("%ld",&us);
    temp=start;
    while(temp!=NULL){
        if(us==temp->serno){
            printf("ENTER YOUR FEEDBACK:");
            scanf("\n");
            scanf("%[^\\n]%*c", temp->feedback);
            temp->fb=1;
            printf("\nTHANK YOU FOR YOUR FEEDBACK");
        }
        temp=temp->next;
    }
    printf("\n PRESS 1 TO GO BACK");
    scanf("%d",&h);
    if(h==1){
        main();
    }
}
void employee(){
    int ch;
    system("cls");
    printf("\n1.VIEW CONNECTIONS");
    printf("\n2.VIEW FEEDBACKS");
    printf("\n3.MAIN MENU");

```

```

while(1){
    printf("\n enter your choice:");
    scanf("%d",&ch);
    switch(ch){
        case 1:
            connections();
            break;
        case 2:
            viewfeedback();
            break;
        case 3:
            main();
            break;
        case 9:
            create();
            break;
        default:printf("\n INVALID CHOICE");
    }
}

void connections(){
    int i=1;
    float bill;
    temp=start;
    system("cls");
    printf("\n S.NO    NAME        SERNO    NO.OF CYLINDERS    BILL AMOUNT
SUBSIDY AMOUNT ");
    while(temp!=NULL){
        printf("\n %d    %s        %ld        %d        %d        %d",i,temp-
>name,temp->serno,temp->cylinder,1124*temp->cylinder,450*temp->cylinder);
        temp=temp->next;
    }
    printf("\n PRESS 1 TO GO BACK");
    scanf("%d",&h);
}

```

```

        if(h==1){
            employee();
        }
    }

void viewfeedback(){
    temp=start;
    system("cls");
    while(temp!=NULL){
        if(temp->fb==1){
            printf("\nSERNO: %ld\n FEEDBACK: %s",temp->serno,temp-
>feedback);
        }
        temp=temp->next;
    }
    printf("\n PRESS 1 TO GO BACK");
    scanf("%d",&h);
    if(h==1){
        employee();
    }
}

```

III.OUTPUT

- 1.NEW CONNECTION ENROLLMENT
- 2.TERMINATE CURRENT CONNECTION
- 3.BOOK A CYLINDER
- 4.FEEDBACK
- 5.EMPLOYEE
- 6.EXIT

Make your choice:1

PLEASE FILL THE FOLLOWING DETAILS FOR NEW CONNECTION:

NAME:sanjay

MOBILE NUMBER:9010994175

HOUSE NO:4-131

STREET:kurmawada

VILLAGE:katlakunta

TOWN:jagtial

CONNECTION SUCCESSFULLY REGISTERED WITH-

SERVICE NO (SERNO) :1

Make your choice:3

ENTER YOUR SERVICE NUMBER:1

ENTER THE NO.OF CYLINDERS REQUIRED5

ENTER 1 TO CONFIRM BOOKING 1

BOOKING SUCCESSFULL.

BHARAT GAS

SERVICE NO :1

NAME :sanjay
H.NO :4-131
ADDRESS :kurmawada,
katlakunta,
jagtial
MOBILE NO :421059583

NO.OF CYLINDERS :5
BASE RATE :3600
CENTRAL GST :90
STATE GST :90
TOTAL BILL :3780

PRESS 1 TO GO BACK 1
1.NEW CONNECTION ENROLLMENT
2.TERMINATE CURRENT CONNECTION
3.BOOK A CYLINDER
4.FEEDBACK
5.EMPLOYEE
6.EXIT

Make your choice:4

ENTER YOUR SERVICE NO:1

ENTER YOUR FEEDBACK:9

THANK YOU FOR YOUR FEEDBACK
PRESS 1 TO GO BACK 1
1.NEW CONNECTION ENROLLMENT
2.TERMINATE CURRENT CONNECTION
3.BOOK A CYLINDER
4.FEEDBACK
5.EMPLOYEE
6.EXIT

Make your choice:2

ENTER YOUR SERVICE NUMBER:1

1.CONIFIRM DELETE

2.SKIP

1

.

THE CONECTION WITH SERVICE NO 1 IS DELETED

PRESS 1 TO GO BACK

CONCLUSION

In conclusion, The GAS BOOKING using a Double linked list in C language is a great data structure for storing of contact information deletion, feedback, and for booking. It allows for fast search times and the ability to store and update contact information quickly. Additionally, the linked list structure allows for easy management of large databases of contacts. The GAS BOOKING implemented using a Double linked list in C language has proved to be an efficient and effective way to store and access contact information. This automated system has the advantage of maintaining the records properly without any manual stress. The program is able to successfully add, delete, booking and feedback for a new connection. It is able to handle large amounts of data efficiently. The double linked list structure is an efficient data structure and has made the program run faster. Hence it becomes easier for the user to automate the property auctioned .it has been proved to be a useful tool for managing contact information.