

**DSA [201ES117] COURSE PROJECT REPORT**

**on**

# ELECTRICITY MANAGEMENT SYSTEM

A Course Work Project Report submitted in partial fulfilment of the requirement for the award of the degree of

**DATA STRUCTURES AND ALGORITHM Computer Science & Engineering**

**Developed By:**

**2103A51489 - R.JAYANTH REDDY**

**Under the guidance of:**

**Dr. A.SHIVA KRISHNA REDDY Associate Professor**

**Submitted To:**

**DEPARTMENT OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE Ananthasagar village, Hasanparthy Mandal, Warangal District – 500 376** [**www.sru.edu.in**](http://www.sru.edu.in/)

***(2022 – 2023)***

**Department of Computer Science and Artificial Intelligence**

0

## CERTIFICATE

This is to certify that the PSP course project report entitled **“ELECTRICITY MANAGEMENT SYSTEM”** is a record of bonafide work carried out by the student **“R.JAYANTH REDDY”** bearing roll number **“2103A51489”** of Computer Science and Artificial Intelligence department during the academic year 2022-23.

**Supervisor**

(Dr. A. Siva Krishna Reddy)

## ACKNOWLEDGEMENTS

First and foremost, we express our sincere thanks for the guidance and encouragement rendered by **A. Shiva Krishna Reddy , Assistant Professor** in the Department of Computer Science & Artificial Intelligence, SR University, Ananthasagar, Warangal District. We extend our gratitude for his advice and guidance during the progress of this course project.

Secondly, We express our sincere thanks to **Dr. M. Shashikala, Associate Professor & Head**, Department of CS & AI, SR University who stood as silent inspiration behind this course project. Our heartfelt thanks for her endorsement and valuable suggestions.

We wish to express our profound thanks to **Dr. R. Archana Reddy, Dean, School of Sciences** for providing necessary facilities to make this course project a success.

We thank all the members of teaching and non-teaching staff members, and also who have assisted us directly or indirectly for successful completion of this course project.

Finally, We would like to express our sincere gratitude to our parents who are constantly encouraging us through-out our lives and for completion of this course project.

***INDEX***

|  |  |
| --- | --- |
| 1. INTRODUCTION | 4 |
| 2. ABSTRACT | 5 |
| 3. PROBLEMSTATEMENT | 6 |
| 4. SYSTEM DESIGN | 7 - 8 |
| 5. MODULES | 9 - 10 |
| 6. HARDWARE  REQUIREMENT | 11 |
| 7. SOFTWARE  REQUIREMENT | 12 |
| 8. OPERATING SYSTEM | 13 |
| 9. CODE | 14 - 36 |
| 10.OUTPUTS | 37- 39 |
| 11.CONCLUSION | 40 |
| 12.REFERENCE | 41 |

**INTRODUCTION**

Electricity Management System Project in C with Source Code – This C Electricity Management System Project is a console-based application written in C. This system is a simple little project that was created in the Embarcadero Dev C++ and compiled with the Dev C++ compiler. The Electricity Management System is a simple console program that does not include any images.

Usually People have to wait for a long duration of time to get a new connection or to terminate the already existing account. But with the help of this you just have to follow a very simple process to pay bills and you need not to wait to a longer duration of time , get a new connection or terminate it and can manage your bills in a easy way.

As, can pay the current bill, get a new connection and can terminate already existing account very easily, Besides this the work load of electricity officers can be reduced to an extinct .

***ABSTRACT***

This project helps the user to pay the electricity bill, get a new connection or terminate the already existing connection in mobile software and bill so we can do payment, The number of wats used will be displayed in menu so that we can pay the bill accordingly.

**The project support following operations:**

▫ Adding new connection

▫ Deleting the existing connections

▫ Giving feedbacks

▫ Proceeding to bill

▫ Access to Employee

This project is implemented in C language using Linked list , loops, arrays, strings, structures and pointers. The program is a MENU driven program which keeps on executing until user selects exit option.

# PROBLEM STATEMENT

* The electrical officers in India especially in rural areas they need to visit each and every house in order to collect electricity bills, to check any issues regarding electric connection to a house and etc.
* But with the help of this application we can save the time of the officers and decrease their efforts . The people will also get benefited by using this application By using this application the user can be able to get a new account/connection or can remove his/her previous existing connection and can pay the electricity bills.

# SYSTEM DESIGN

In this phase, a logical system is built which fulfils the given requirements. Design phase of software development deals with transforming the clients’s requirements into a logically working system. Normally, design is performed in the following in the following two steps:

1. **Primary Design Phase:**

In this phase, the system is designed at block level. The blocks are created on the basis of analysis done in the problem identification phase. Different blocks are created for different functions emphasis is put on minimising the information flow between blocks. Thus, all activities which require more interaction are kept in one block.

1. **Secondary Design Phase:**

In the secondary phase the detailed design of every block is performed.

The general tasks involved in the design process are the following:

1. Design various blocks for overall system processes.
2. Design smaller, compact and workable modules in each block.
3. Design various database structures.
4. Specify details of programs to achieve desired functionality.
5. Design the form of inputs, and outputs of the system.
6. Perform documentation of the design.
7. System reviews

**User Interface Design**

User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs. The overall flow of screens and messages is called a dialogue.

The following steps are various guidelines for User Interface Design:

1. The system user should always be aware of what to do next.
2. The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area.
3. Message, instructions or information should be displayed long enough to allow the system user to read them.
4. Use display attributes sparingly.
5. Default values for fields and answers to be entered by the user should be specified.
6. A user should not be allowed to proceed without correcting an error.
7. The system user should never get an operating system message or fatal error

# 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 (do-while) is used with condition. In this program sting functions are used. The application asks the person who runs the program.

In this application five modules are used:

* Create() :

This function is used to create a new node

* Bill() :

This function is used to show the total bill to user on his service number

* Newenrol() :

This function is used to enrol a new account to user .

* Feedback() :

This function allows the user to give any feedbacks on this application .

* Viewfeedback():

This function allows the admin to view the feedbacks given by users on this application .

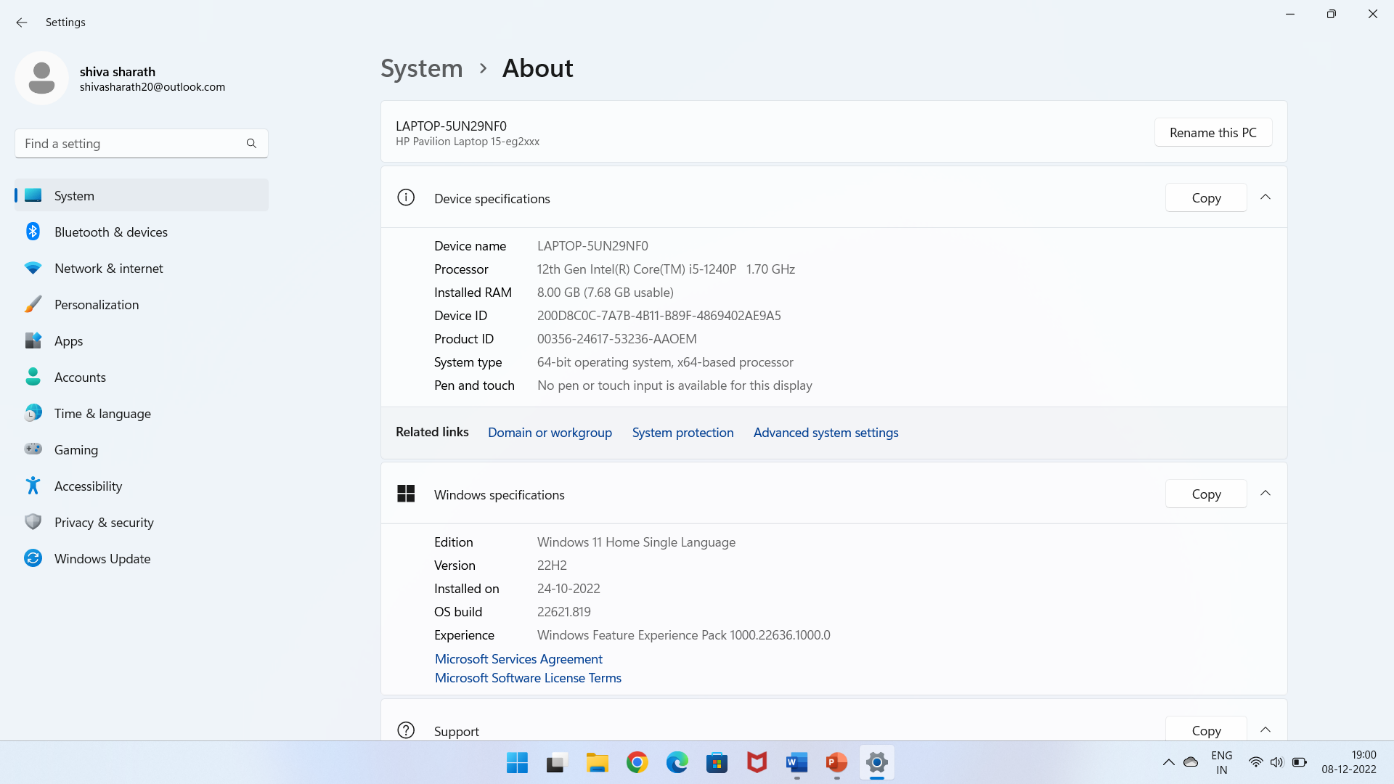
* Employee():

This function can be used by the employees of Gas stations to view the connections and feedbacks of users .

## DOUBLE LINKED LIST

* A **Double linked list** is a type of linked list that is *bidirectional*, that is, it can be traversed in two directions from head to the last node (tail) and in last node to head (reverse) also.
* Each element in a linked list is called a **node**. A single node contains *data* and two pointers to the *next* node previous node which helps in maintaining the structure of the list.
* The first node is called the **head**; it points to the first node of the list and helps us access every other element in the list. The last node, also sometimes called the **tail**, points to *NULL* which helps us in determining when the list ends.

## HARDWARE



# Software : Embarcadero Dev C++



Dev-C++ is a fully featured graphical IDE (Integrated Development Environment) that uses the MinGw compiler system to create Windows as well as Console based C/C++ applications. It can also be used with any other GCC-based compiler like Cygwin. Dev-C++ is free software and is distributed under the GNU General Public License.

Embarcadero Dev-C++ is a new and improved fork (sponsored by Embarcadero) of Bloodshed Dev-C++ and Orwell Dev-C++.

# OPERATING SYSTEM

Windows 11 is the latest major release of Microsoft's Windows NT operating system, released in October 2021. It is a free upgrade to its predecessor, Windows 10 (2015), available for any Windows 10 devices that meet the new Windows 11 system requirements.

Windows 11 features major changes to the Windows shell influenced by the canceled Windows 10X, including a redesigned Start menu, the replacement of its "live tiles" with a separate "Widgets" panel on the taskbar, the ability to create tiled sets of windows that can be minimized and restored from the taskbar as a group, and new gaming technologies inherited from Xbox Series X and Series S such as Auto HDR and DirectStorage on compatible hardware. Internet Explorer (IE) has been replaced by the Chromium-based Microsoft Edge as the default web browser like its predecessor, Windows 10, and Microsoft Teams is integrated into the Windows shell. Microsoft also announced plans to allow more flexibility in software that can be distributed via Microsoft Store, and to support Android apps on Windows 11 (including a partnership with Amazon to make its app store available for the function).

# PROGRAM

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct Node{

long unsrno;

long serno;

char name[100];

char hmno[100];

char street[100];

char village[100];

char town[100];

char feedback[1000];

int fb;

long units;

int pay;

struct Node \*prev;

struct Node \*next;

};

struct Node \*start;

struct Node \*temp,\*newnode,\*ptr;

int h;

long a=16362700;

long b=1231409800;

void bill();

void newenrol();

void terminate();

void feedback();

void viewfeedback();

void create();

void reading();

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.PAYMENT AND BILLING");

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==1435){

employee();

}

else{

printf("\nERROR! INVALID EMLOYEE ID");

}

break;

case 6:

exit(0);

break;

default:

printf("\n Invalid choice");

break;

}

}

}

void create()

{

struct Node \*newnode,\*temp;

int total,index,item,h;

system("cls");

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(" HOUSE NO:");

scanf("%s",temp->hmno);

printf(" STREET:");

scanf("%s",temp->street);

printf(" VILLAGE:");

scanf("%s",temp->village);

printf(" SUB-STATION:");

scanf("%s",temp->town);

printf(" UNIQUE SERVICE NO:");

scanf("%ld",temp->unsrno);

printf(" SERVICE NO:");

scanf("%ld",temp->serno);

temp->pay=1;

temp->fb=0;

temp->prev=NULL;

temp->next=NULL;

start = temp;

printf("\nCONNECTION SUCCESSFULLY REGISTERED WITH-");

printf("\nUNIQUE SERVICE NO (USCNO):%ld",temp->unsrno);

printf("\nSERVICE NO (SCNO) :%ld",temp->serno);

for(index=2;index<=total;index++)

{

newnode = (struct Node\*)malloc(sizeof(struct Node));

printf("\n Enter the person details:");

printf("\n NAME:");

scanf("%s",newnode->name);

printf(" HOUSE NO:");

scanf("%s",newnode->hmno);

printf(" STREET:");

scanf("%s",newnode->street);

printf(" VILLAGE:");

scanf("%s",newnode->village);

printf(" SUB-STATION:");

scanf("%s",newnode->town);

printf(" UNIQUE SERVICE NO:");

scanf("%ld",newnode->unsrno);

printf(" SERVICE NO:");

scanf("%ld",newnode->serno);

newnode->pay=1;

newnode->fb=0;

newnode->next = NULL;

newnode->prev=temp;

temp->next = newnode;

temp=newnode;

printf("\nCONNECTION SUCCESSFULLY REGISTERED WITH-");

printf("\nUNIQUE SERVICE NO (USCNO):%ld",newnode->unsrno);

printf("\nSERVICE NO (SCNO) :%ld",newnode->serno);

}

}

}

else

{

printf("\n CONNECTIONS ALREADY REGISTERED");

}

printf("\n PRESS 1 TO GO BACK");

scanf("%d",&h);

if(h==1){

employee();

}

}

void bill()

{

struct Node \*temp;

long int us;

float bill;

int a,i;

if(start==NULL)

{

printf("\n error! no enrolled connections");

}

else

{

printf("ENTER YOUR UNIQUE SERVICE NUMBER:");

scanf("%ld",&us);

temp=start;

system("cls");

while(temp!=NULL)

{

if(us==temp->unsrno){

if(temp->pay==0){

if(temp->units<200){

bill=temp->units\*1.20;

}

else if(temp->units>=200||temp->units<500){

bill=temp->units\*1.50;

}

else{

bill=temp->units\*2;

}

printf(" TSNPDCL ");

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

printf("\nUSCNO :%ld",temp->unsrno);

printf("\nNAME :%s",temp->name);

printf("\nSCNO :%ld",temp->serno);

printf("\nADDR :%s",temp->hmno);

printf("\nSUB STN :%s",temp->town);

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

printf("\nNO.OF UNITS :%ld",temp->units);

printf("\nENERGY CHARGES :%f",bill);

printf("\nFIXED CHARGES :20");

printf("\nCUST CHARGES :40");

printf("\nTOTAL BILL :%f",bill+60);

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

printf("\n1.PAY NOW\n2.SKIP\n");

scanf("%d",&a);

if(a==1){

for(i=0;i<7;i++){

printf(".\t");

sleep(1);

}

printf("\n BILL PAID SUCCESSFULLY");

temp->pay=1;

}

}

else{

printf("\n BILL IS ALREADY PAID \n NO NEW BILL AS OF NOW");

}

}

temp=temp->next;

}

printf("\n PRESS 1 TO GO BACK");

scanf("%d",&h);

if(h==1){

main();

}

}

}

void newenrol()

{

struct Node \*temp;

struct Node \*newnode;

int item;

newnode=(struct Node \*)malloc(sizeof(struct Node));

newnode->unsrno=a;

newnode->serno=b;

printf("\n PLEASE FILL THE FOLLOWING DETAILS FOR NEW CONNECTION:");

printf("\n NAME:");

scanf("\n");

scanf("%[^\n]%\*c",newnode->name);

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(" SUB-STATION:");

scanf("\n");

scanf("%[^\n]%\*c",newnode->town);

newnode->pay=1;

newnode->fb=0;

newnode->next=NULL;

newnode->prev=NULL;

if(start==NULL){

start=newnode;

newnode->unsrno=newnode->unsrno+1;

newnode->serno=newnode->serno+1;

printf("\nCONNECTION SUCCESSFULLY REGISTERED WITH-");

printf("\nUNIQUE SERVICE NO (USCNO):%ld",newnode->unsrno);

printf("\nSERVICE NO (SCNO) :%ld",newnode->serno);

a++;

b++;

}

else

{

temp=start;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=newnode;

newnode->prev=temp;

newnode->unsrno=newnode->unsrno+1;

newnode->serno=newnode->serno+1;

printf("\nCONNECTION SUCCESSFULLY REGISTERED WITH-");

printf("\nUNIQUE SERVICE NO (USCNO):%ld",newnode->unsrno);

printf("\nSERVICE NO (SCNO) :%ld",newnode->serno);

a++;

b++;

}

}

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 UNIQUE SERVICE NUMBER:");

scanf("%ld",&us);

temp=start;

while(temp!=NULL)

{

i++;

temp=temp->next;

}

temp=start;

while(temp!=NULL)

{

if(us==temp->unsrno){

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 UNIQUE SERVICE NO %ld IS DELETED",start->unsrno);

start=NULL;

}

}

else{

printf("\n1.PAY NOW\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 UNIQUE SERVICE NO %ld IS DELETED",start->unsrno);

//start = temp->next;

start = start->next;

start->prev=NULL;

free(temp);

}

}

}

else if(loc==i&&loc!=1){

printf("\n1.PAY NOW\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 UNIQUE SERVICE NO %ld IS DELETED",temp->unsrno);

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.PAY NOW\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 UNIQUE SERVICE NO %d IS DELETED",temp->unsrno);

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 UNIQUE SERVICE NO:");

scanf("%ld",&us);

temp=start;

while(temp!=NULL){

if(us==temp->unsrno){

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.ENTER THE BILLS");

printf("\n3. VIEW FEEDBACKS");

printf("\n4.MAIN MENU");

while(1){

printf("\n enter your choice:");

scanf("%d",&ch);

switch(ch){

case 1:

connections();

break;

case 2:

reading();

break;

case 3:

viewfeedback();

break;

case 4:

main();

break;

case 9:

create();

break;

default:printf("\n INVALID CHOICE");

}

}

}

void reading(){

struct Node \*temp;

temp=start;

if(temp==NULL){

printf("NO CONNECTIONS ENROLLED");

}

else{

system("cls");

while(temp!=NULL){

printf("ENTER NO.OF UNITS USED BY UNIQUE SERVICE NO %d:",temp->unsrno);

scanf("%ld",&temp->units);

temp->pay=0;

temp=temp->next;

}

printf("\n PRESS 1 TO GO BACK");

scanf("%d",&h);

if(h==1){

employee();

}

}

}

void connections(){

int i=1;

float bill;

temp=start;

system("cls");

printf("\n S.NO NAME UNSCNO BILL AMOUNT PAYMENT ");

while(temp!=NULL){

if(temp->units<200){

bill=temp->units\*1.20+60;

}

else if(temp->units>=200||temp->units<500){

bill=temp->units\*1.50+60;

}

else{

bill=temp->units\*2+60;

}

if(temp->pay==1){

printf("\n %d %s %ld %f PAID",i,temp->name,temp->unsrno,bill);

}

else{

printf("\n %d %s %ld %f NOT PAID",i,temp->name,temp->unsrno,bill);

}

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("\nUSCNO: %ld\n FEEDBACK: %s",temp->unsrno,temp->feedback);

}

temp=temp->next;

}

printf("\n PRESS 1 TO GO BACK");

scanf("%d",&h);

if(h==1){

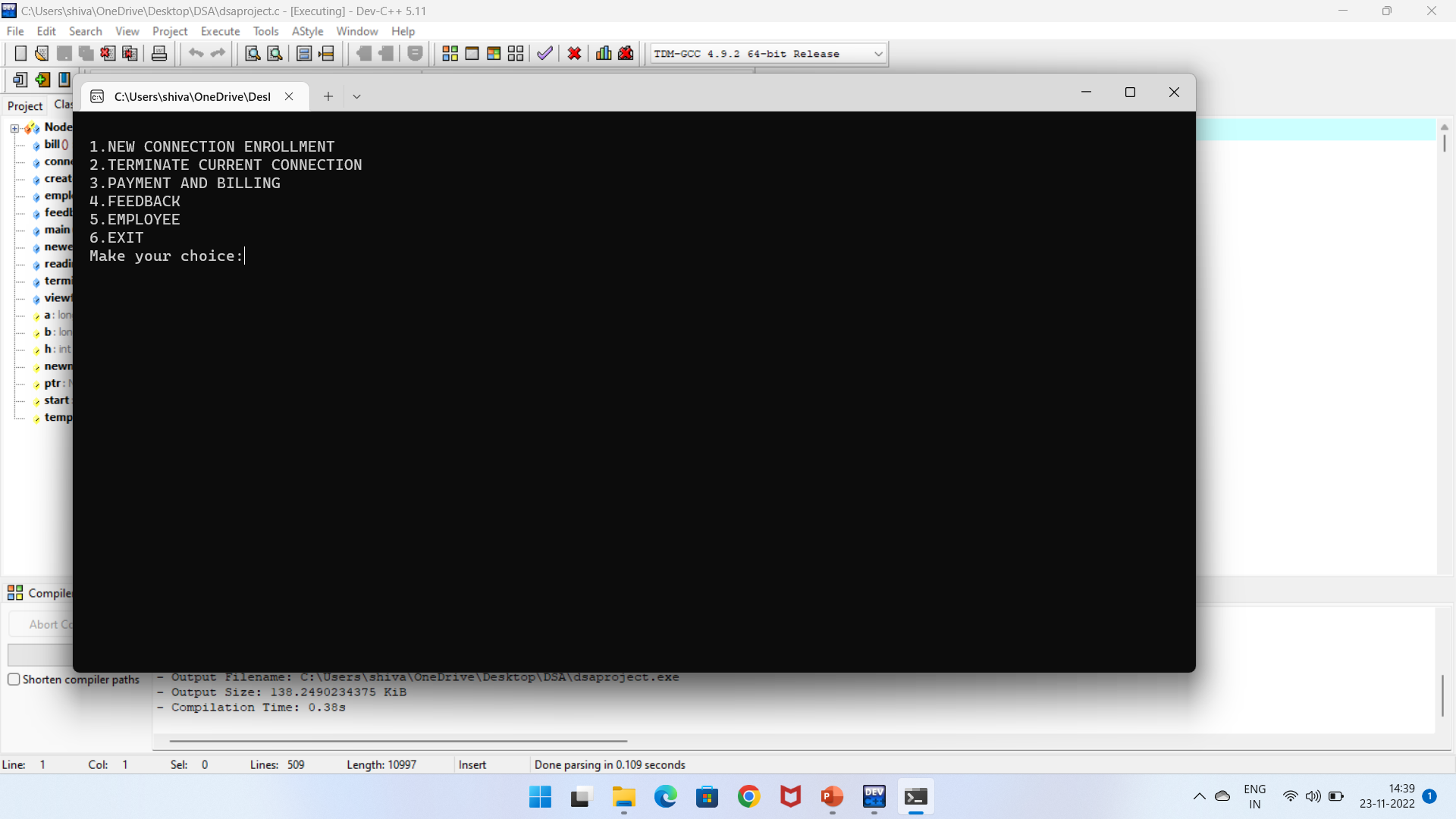
employee();

}

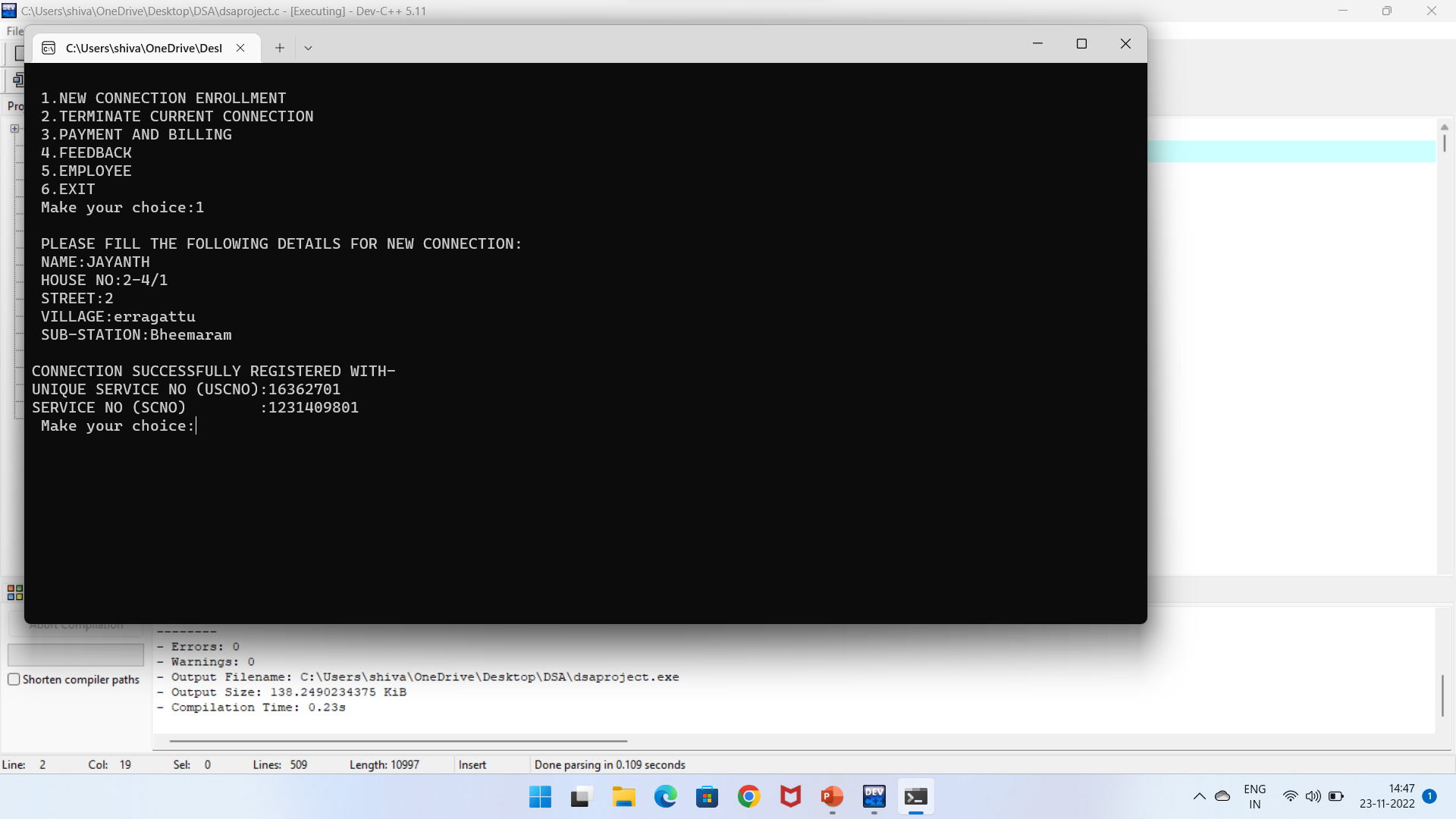
}

# OUTPUTS

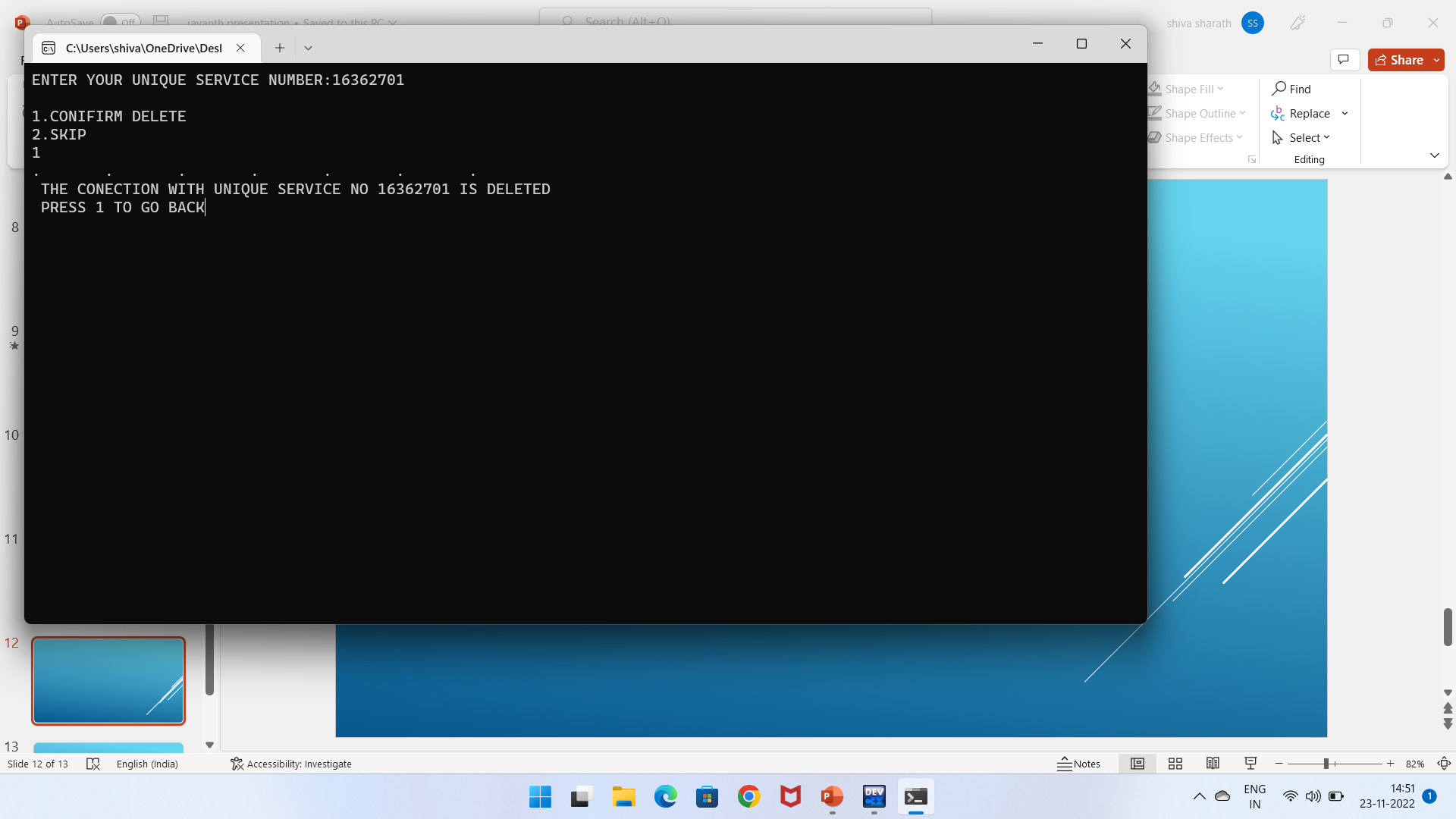
**1. HOMEPAGE:**

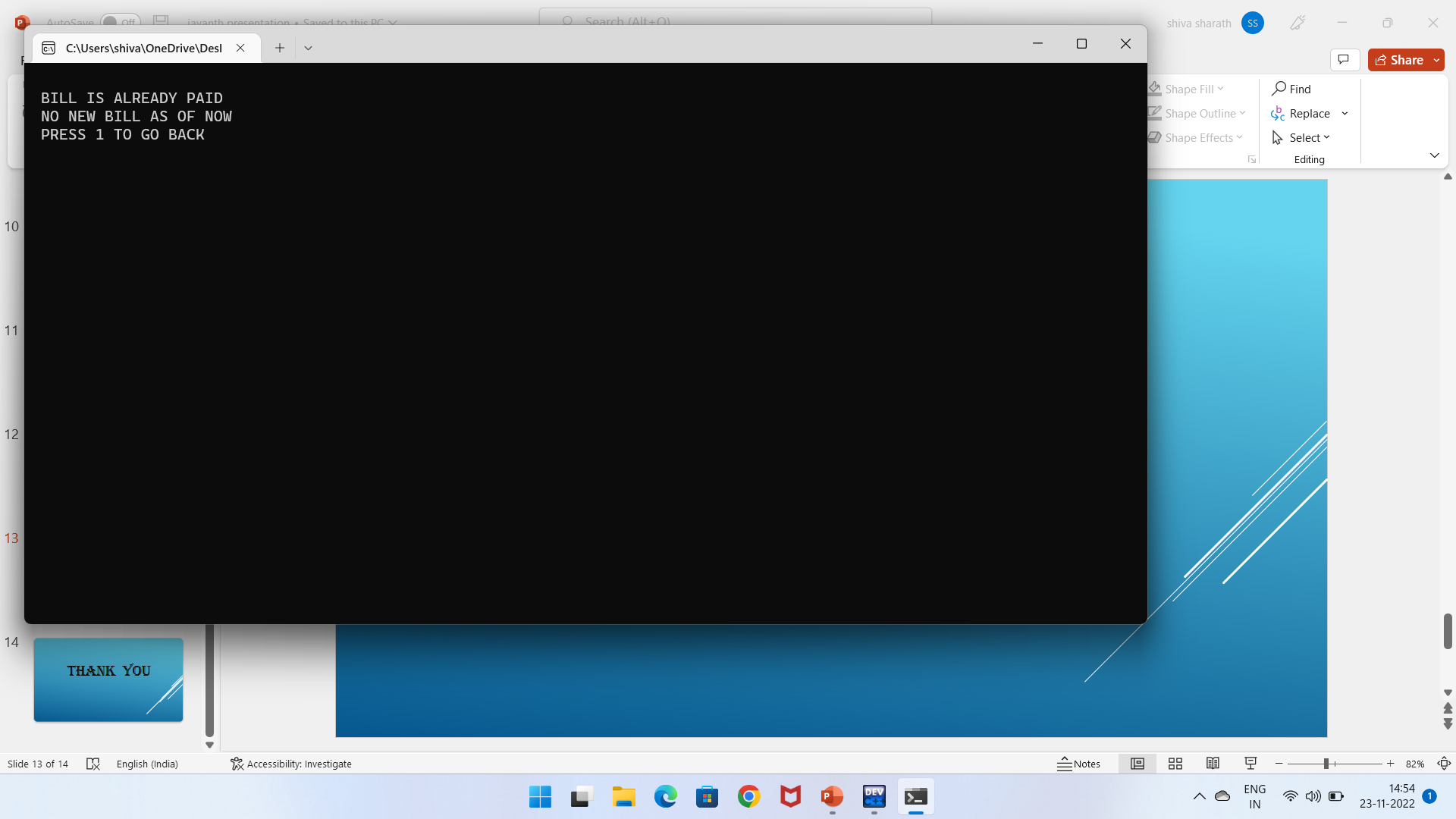


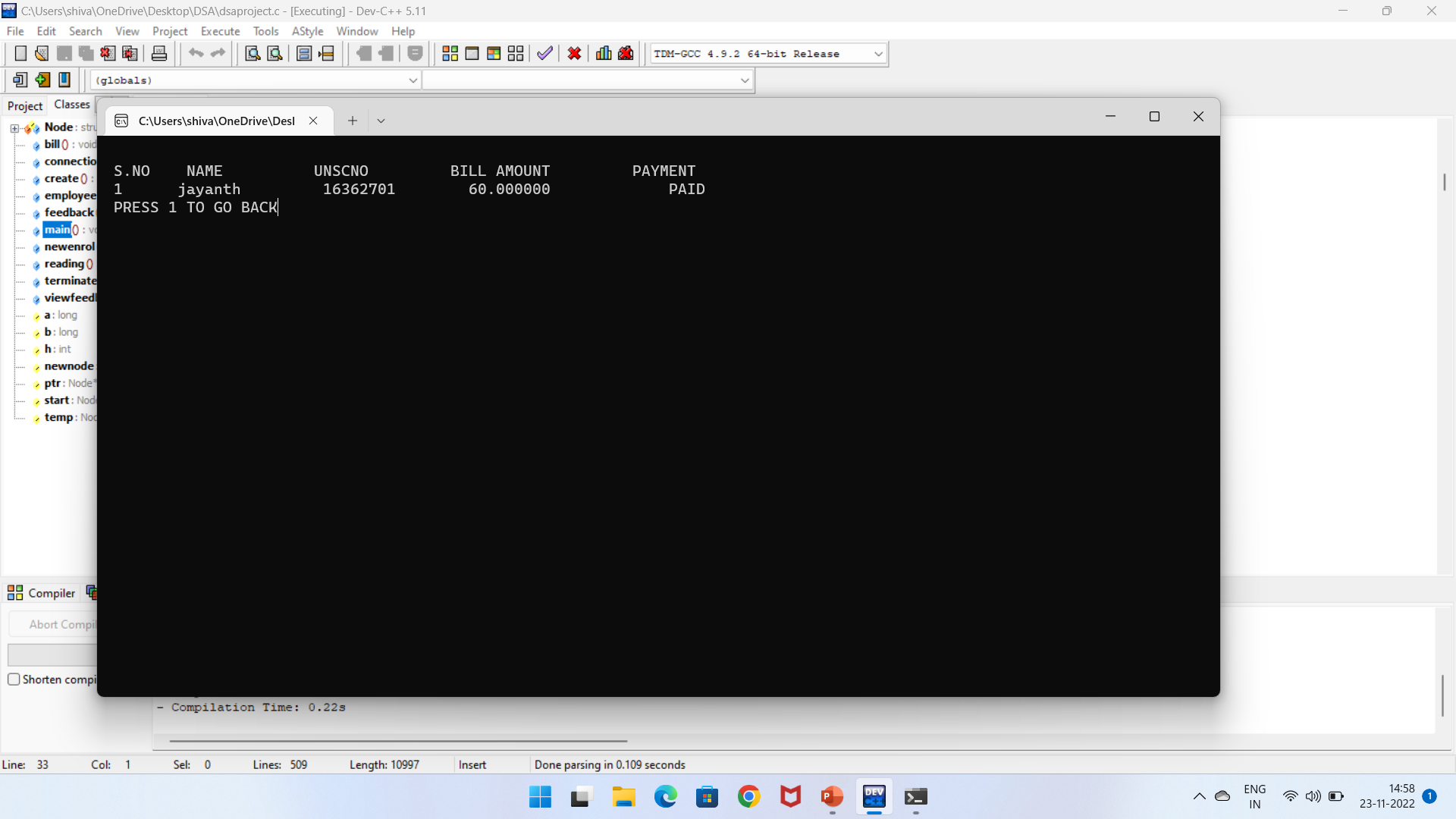
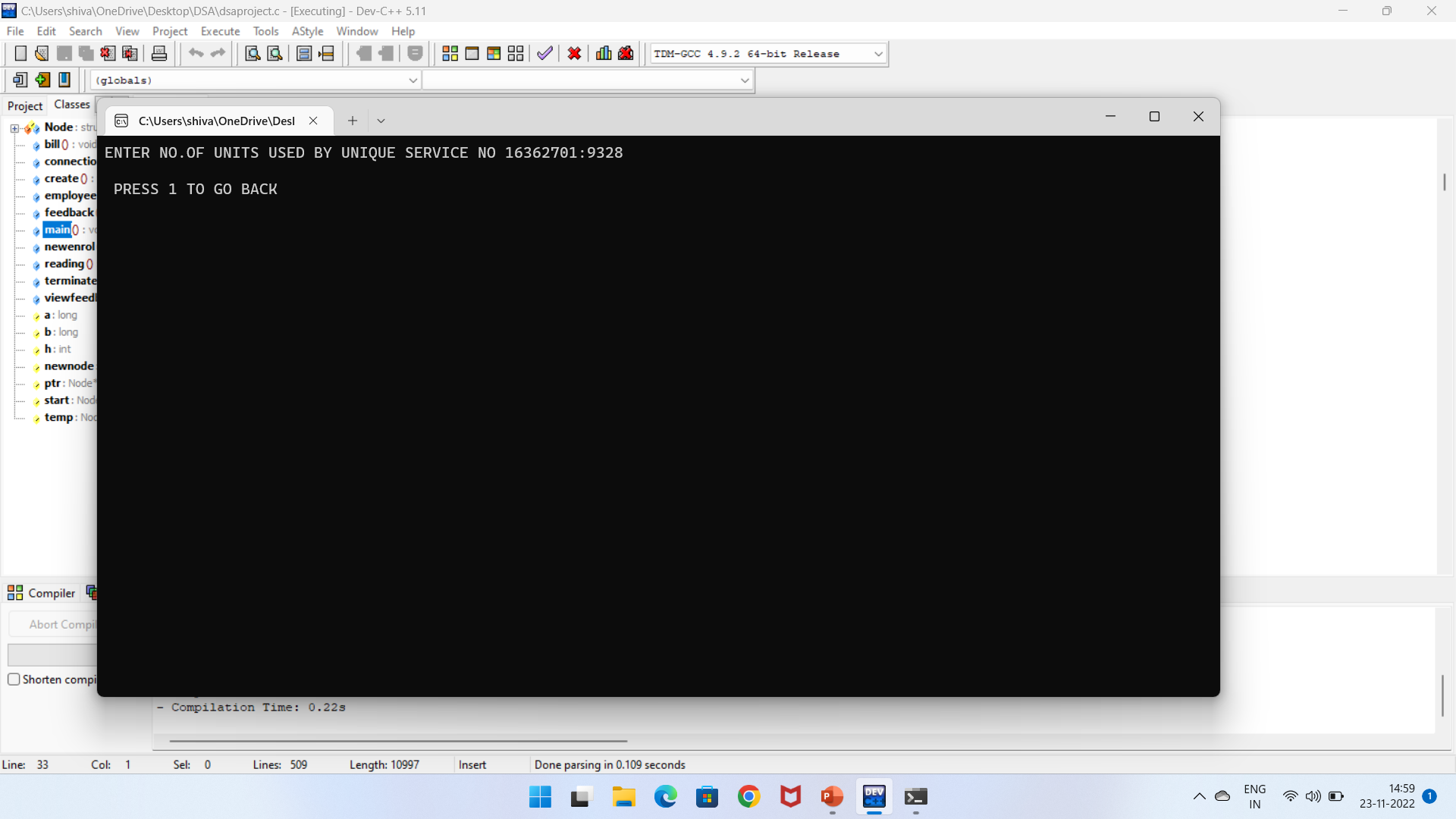
**2.New Enrolment:**

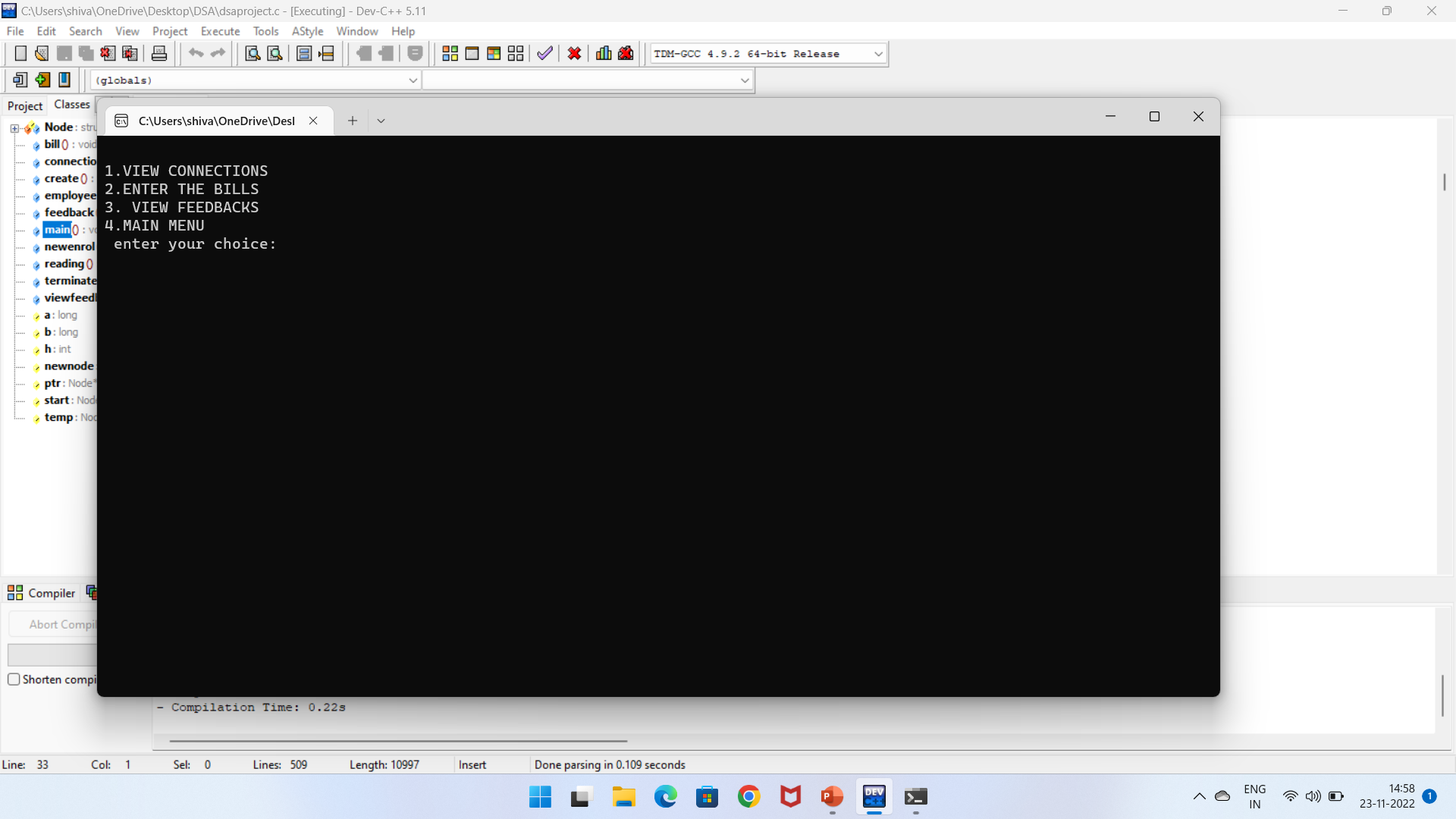


**3.Terminate connection:**

****

4.Payment and Billing: 

5.Employee: 



# CONLUSION

* **This project is efficient in maintaining customer details and can easily performs operations on customer’s records and also works to handles the information of the electrical connection/electricity supply available.**
* **This project also reduces the work load the electric officers and helps them to reduce the efforts.**
* The project explores various features of C-programming language mainly like functions, loops, structures, linked lists and Arrays.

# REFERENCE

**BOOKS:-**

C : The Complete reference

Let Us C.

**VIDEOS*:-***

<https://youtu.be/aTVSEEQXs-Y>

<https://youtu.be/mCTSzm1YcfM>

<https://youtu.be/y9UJn4fqqWk>

<https://www.youtube.com/watch?v=eGnlKPCkAFY>