**DAYANANDA SAGAR UNIVERSITY**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SCHOOL OF ENGINEERING**

**DAYANANDA SAGAR UNIVERSITY**

**KUDLU GATE**

**BANGALORE - 560068**

****

**MINI PROJECT REPORT**

***ON***

**“EMPLOYEE MANAGEMENT SYSTEM”**

**SUBMITTED TO THE 3rd SEMESTER DATA STRUCTURE AND APPLICATIONS LABORATORY**

**BACHELOR OF TECHNOLOGY**

***IN***

**COMPUTER SCIENCE & ENGINEERING**

***Submitted by***

KURUVA DIVYA SREE– (ENG18CS0144)

LAKSHMI POOJITHA–(ENG18CS0150)

L K SABARISH-(ENG18CS0145)

KEVIN PAUL-(ENG18CS0134)

***Under the supervision of***

**Prof. Ankita Singhai**

**DAYANANDA SAGAR UNIVERSITY**

**School of Engineering, Kudlu Gate, Bangalore-560068**

****

**CERTIFICATE**

***This is to certify that Mr./Ms.*** KURUVA DIVYA SREE  ***bearing USN***

***ENG18CS0144 has satisfactorily completed his/her Mini Project as prescribed by the University for the III semester B.Tech. programme in Computer Science & Engineering during the year 2019-2020 at the School of Engineering, Dayananda Sagar University., Bangalore.***

Date: 26-11-2019

Signature of the faculty in-charge

|  |  |
| --- | --- |
| Max Marks | Marks Obtained |
|  |  |

Signature of Chairman

Department of Computer Science & Engineering

**DECLARATION**

We hereby declare that the work presented in this mini project entitled -“ EMPLOYEE MANAGEMENT SYSTEM “, has been carried out by us and it has not been submitted for the award of any degree,diploma or the mini project of any other college or university.

KURUVA DIVYASREE (ENG18CS0144)

LAKSHMI POOJITHA (ENG18CS0150)

L K SABARISH (ENG18CS145)

KEVIN PAUL (ENG18CS134)

**ACKNOWLEDGEMENT**

We are pleased to acknowledge **Prof. Ankita Singhai** for her invaluable guidance, support, motivation and patience during the course of this mini- project work.

We extend our sincere thanks to our **Chairman Dr. Banga M.K** who continuously helped throughout the project and without his guidance, this project would have been an uphill task.

We have received a great deal of guidance and co-operation from my friends and I wish to thank one and all that have directly or indirectly helped me in the successful completion of this mini-project work.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **Contents** | **Page no** |
| Abstract | **6** |
| Introduction | **7** |
| Problem Statement | **8** |
| Objective | **9** |
| Software And Hardware Requirements | 10 |
| Methodologies | **11-18** |
| Result | **19-21** |
| Conclusion | **22** |
| Future enhancement | **23** |
| References | **24** |

**ABSTRACT**

As we are beginners and have no practical experience in the field of software development and moreover the employee management system is a very vast network, we limit the scope if our project by computerizing the following fields of the employee management system :

1. Creating a record of a employee.
2. Displaying the created record.
3. Searching for employee record.
4. Deleting the employee records.

In the software developed, there will be separate functions for each of the above points so that there is sample scope for adding more features in the near future.

**INTRODUCTION**

Employee Management system is an application that enables users to create and store Employee Records. This application is helpful to department of the organization which maintains data of employees related to an organization.

C is a platform independent language. Its created applications can be used on a standalone machine as well as on distributed network. More over applications developed in C can be extended to Internet based applications. Thus C was chosen as background to design this application.

**PROBLEM STATEMENT**

An employee is a very important part of the any organizations. Without the employees, there would not be any proper functioning of the organization. Employees are the backbone of any company. The employee management system is an application that will track the records all the employees of the organization. This application will contain the details like name, designation, salary information, attendance details, leave details, client details, project details etc. The performance of the employees can also be kept track of through this application. This application can contain the administrator who can post the announcements, holidays, birthdays and notes. This will be one of the interesting projects that one can work on and implement in real time world.

The “Employee Management System” has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some case, reduce the hardship faced by the existing system to create a employee management System using ‘C’ Language involving concepts of Searching, deleting and inserting. This software will contain all the necessary information regarding employee details.

**OBJECTIVE OF THE PROJECT**

In this world of growing technologies everything has been computerized. With large number of work opportunities the Human workforce has increased. Thus there is a need of a system which can handle the data of such a large number of Employees in an organization. This project simplifies the task of maintain records because of its user friendly nature.

This project aims to simplify the task of maintaining records of a company,schools,college etc.To develop an well-designed database to store employee information.The objective of this project is to provide a comprehensive approach towards the management of employee information.

**SOFTWARE AND HARDWARE REQUIREMENTS**

**Software requirements :**

* Programming language C
* Online GDB compiler

**Hardware requirements :**

* PC or Laptop
  + RAM min- 512 MB
  + Hard disk space-500MB
  + Keyboard
  + High resolution screen

**METHODOLOGY**

**Data structures used**: Linked Lists

Like arrays, Linked List is a linear data structure. Unlike arrays, linked list elements are not stored at a contiguous location; the elements are linked using pointers.

[](https://media.geeksforgeeks.org/wp-content/cdn-uploads/gq/2013/03/Linkedlist.png)

**Why Linked List?**

Arrays can be used to store linear data of similar types, but arrays have the following limitations.  
**1)** The size of the arrays is fixed: So we must know the upper limit on the number of elements in advance. Also, generally, the allocated memory is equal to the upper limit irrespective of age.   
**2)** Inserting a new element in an array of elements is expensive because the room has to be created for the new elements and to create room existing elements have to be shifted.

**Advantages over arrays**  
**1)** Dynamic size  
**2)** Ease of insertion/deletion

Arrays and Linked Lists both are linear data structures, but they both have some advantages and disadvantages over each other.

One advantage of the linked list is that elements can be added to it indefinitely, while an array will eventually get filled or have to be resized (a costly operation that isn't always possible).

Elements are also easily removed from a linked list whereas removing elements from an array leaves empty spaces that are a waste of computer memory.

**FLOWCHART DEPICTING EMPLOYEE MANAGEMENT SYSTEM**

**CODE**

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<string.h>**

**struct node \*createnode (struct node \*);**

**void display (struct node \*);**

**void search (struct node \*);**

**void delete (struct node \*);**

**struct node**

**{**

**char name[10], des[100];**

**int age;**

**float salary;**

**struct node \*ptr;**

**};**

**int main ()**

**{**

**struct node \*head;**

**int b, i;**

**head = NULL;**

**while (1)**

**{**

**printf("\nEnter the value\n1-> to enter the employee details\n2->to display the results\n3->to search an element\n4->to delete:\n");**

**scanf ("%d", &b);**

**switch (b)**

**{**

**case (1):**

**printf("\nEnter the number of employee details you would like to enter\n");**

**scanf ("%d", &i);**

**while (i > 0)**

**{**

**head = createnode (head);**

**i--;**

**}**

**break;**

**case (2):**

**display (head);**

**break;**

**case (3):**

**search (head);**

**break;**

**case (4):**

**delete (head);**

**break;**

**}**

**}**

**}**

**void display (struct node \*head)**

**{**

**if (head == NULL)**

**{**

**printf ("\nThe node is yet to be displayed \n");**

**}**

**else**

**{ printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*EMPLOYEE DETAILS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");**

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

**printf("\nName\t\t designation\t\tsalary\t\tage\t\t");**

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

**while (head != NULL)**

**{**

**printf ("\n%4s\t", head->name);**

**printf ("\t%11s\t", head->des);**

**printf ("\t%6.2f\t", head->salary);**

**printf ("\t%3d\t", head->age);**

**head = head->ptr;**

**}**

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

**}**

**}**

**struct node \* createnode (struct node \*head)**

**{**

**struct node \*newnode;**

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

**printf ("\nEnter the employee name \n");**

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

**printf ("\nEnter the employee's designation \n");**

**scanf ("%s", newnode->des);**

**printf ("\nEnter the salary of the employee \n");**

**scanf ("%f", &newnode->salary);**

**printf ("\nEnter the age of the employee\n");**

**scanf ("%d", &newnode->age);**

**if (newnode == NULL)**

**{**

**printf ("\n Enter the newnode \n");**

**newnode->ptr = NULL;**

**}**

**else**

**{**

**newnode->ptr = head;**

**}**

**return newnode;**

**}**

**void search (struct node \*head)**

**{**

**char ch[100];**

**struct node \*temp=head;**

**struct node \*loc=NULL;**

**printf ("\nEnter the search string\n");**

**scanf ("%s", ch);**

**while (temp!= NULL)**

**{**

**if (strcmp (ch, temp->name) == 0)**

**{**

**printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*EMPLOYEE DETAILS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");**

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

**printf("\nName\t\t designation\t\tsalary\t\tage\t\t");**

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

**printf ("\n%4s\t", temp->name);**

**printf ("\t%11s\t", temp->des);**

**printf ("\t%6.2f\t", temp->salary);**

**printf ("\t%3d\t", temp->age);**

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

**loc=temp;**

**break;**

**}**

**else**

**{**

**temp=temp->ptr;**

**}**

**}**

**if(loc==NULL)**

**{**

**printf("element not matched\n");**

**}**

**}**

**void delete (struct node \*head)**

**{**

**int posi, count = 1;**

**struct node \*locp;**

**struct node \*loc;**

**printf ("enter the position to delete\n");**

**scanf ("%d", &posi);**

**if (head == NULL)**

**printf ("list is empty");**

**else**

**{**

**if (posi == 1)**

**{**

**locp = NULL;**

**loc = head;**

**}**

**else**

**{**

**locp = head;**

**loc = head->ptr;**

**count = 2;**

**while (count != posi)**

**{**

**locp = loc;**

**loc = loc->ptr;**

**count++;**

**}**

**if (locp == NULL)**

**head = loc->ptr;**

**else**

**locp->ptr = loc->ptr;**

**}**

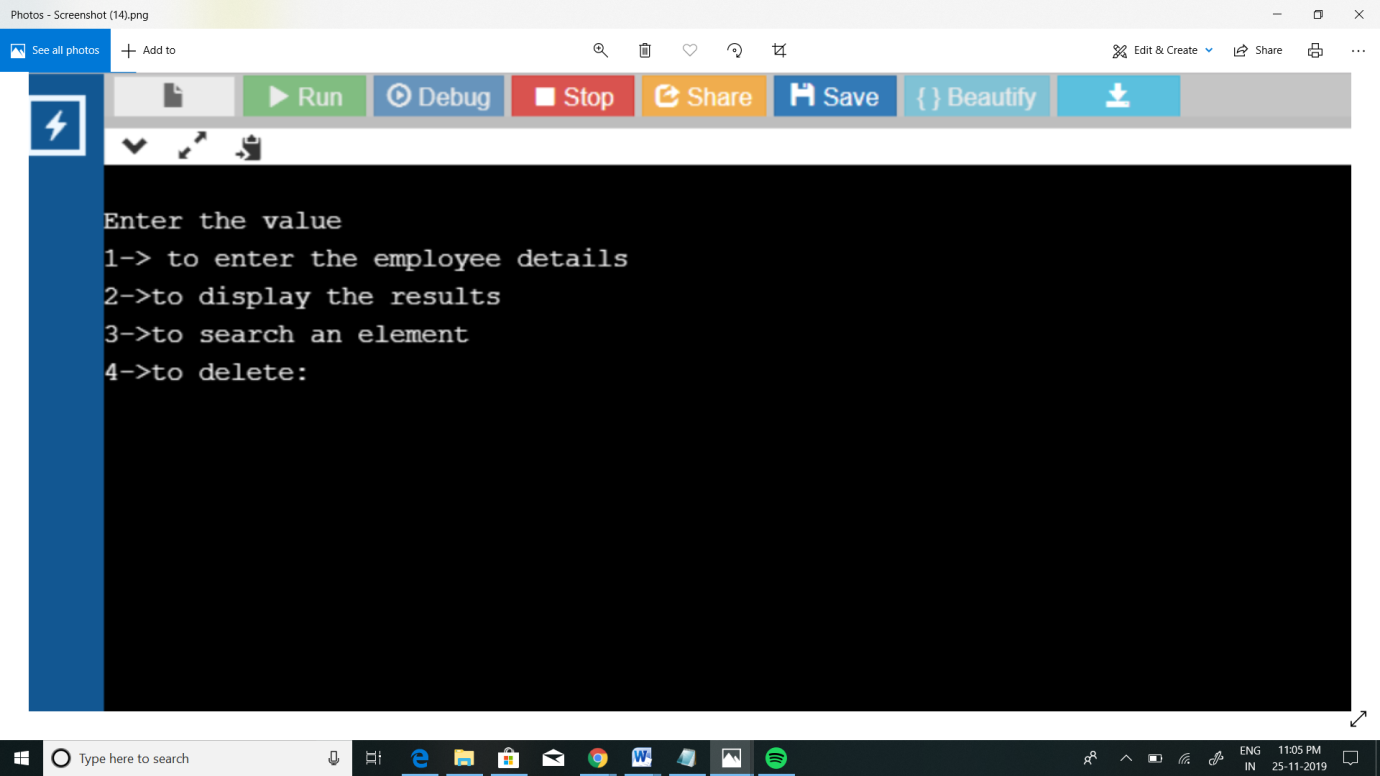
**free (loc);**

**printf ("the node is deleted\n");**

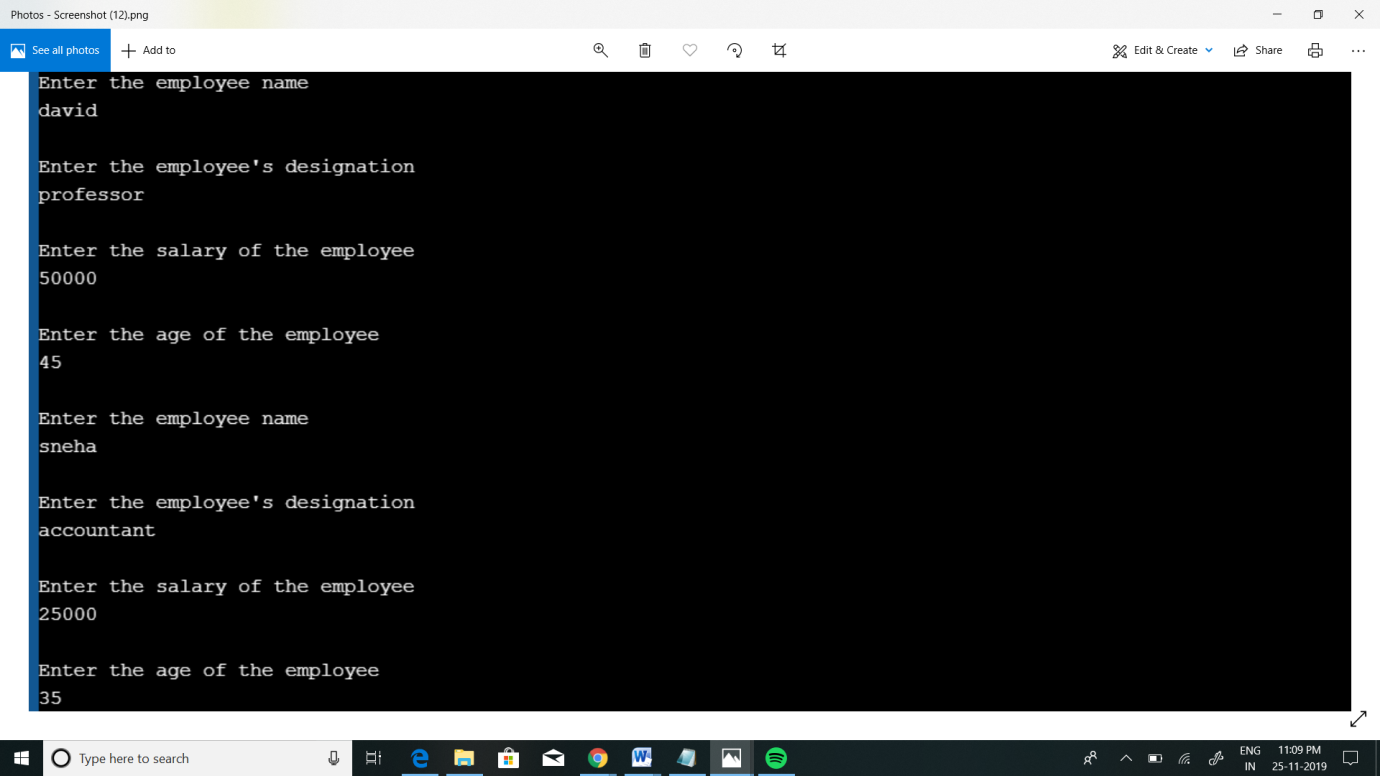
**}**

**}**

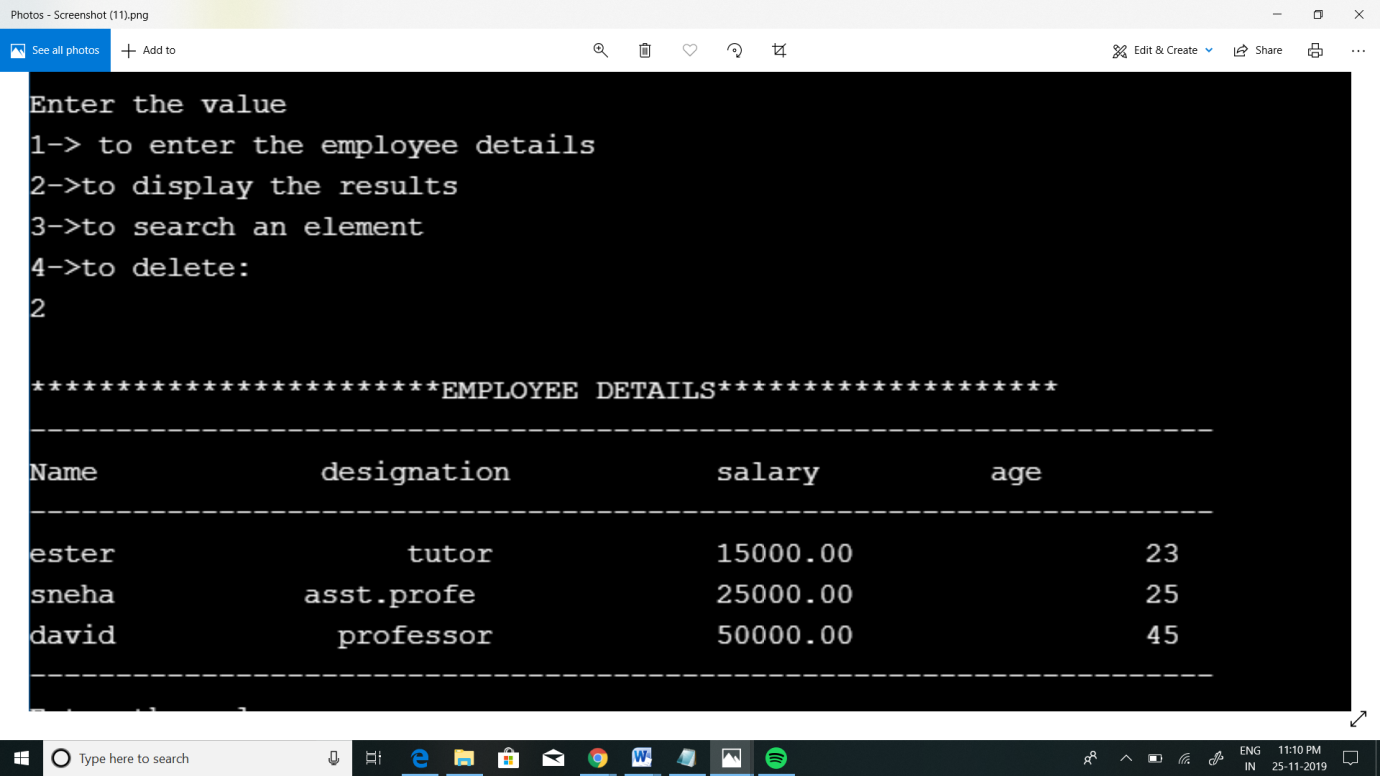
**RESULT**

****

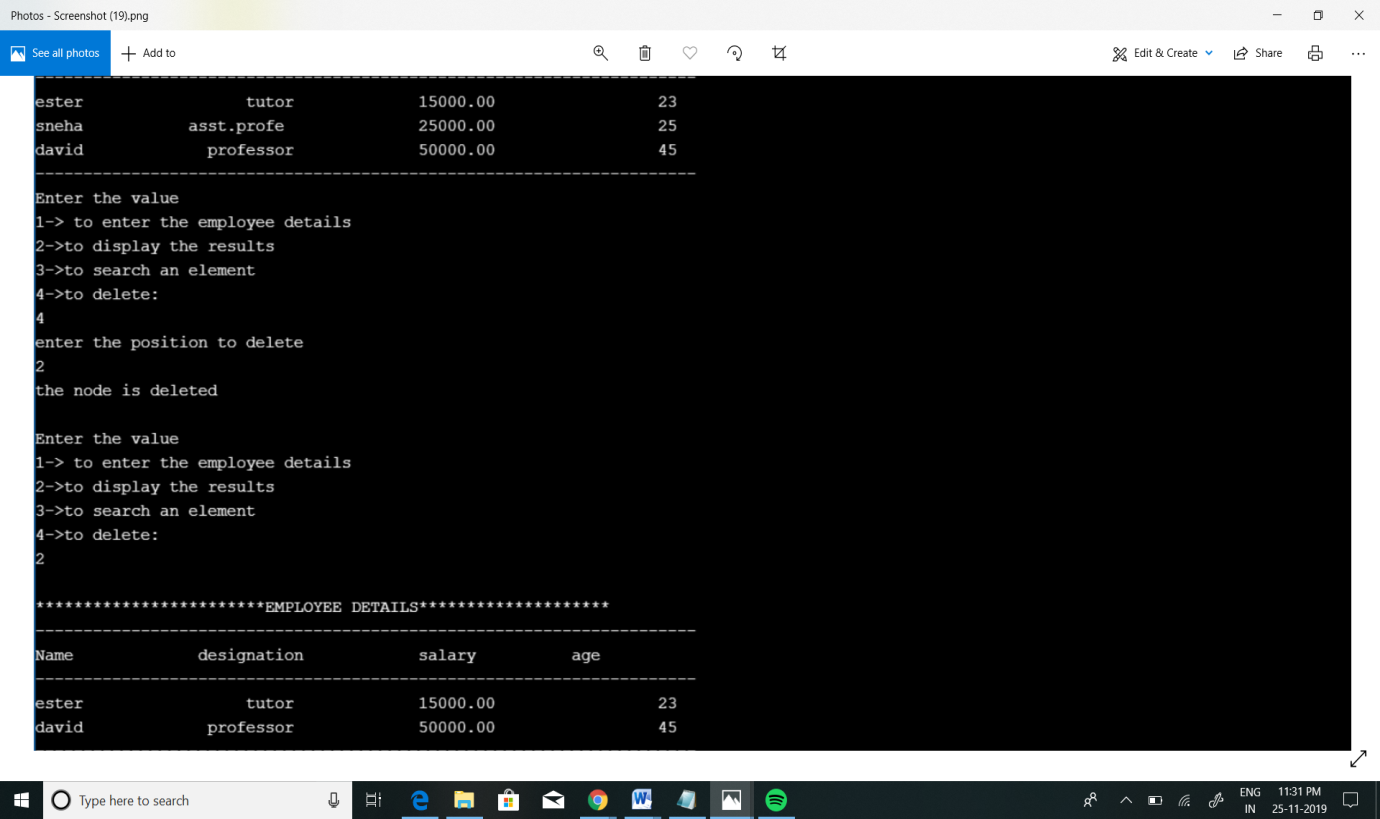
**Fig**.1.main function



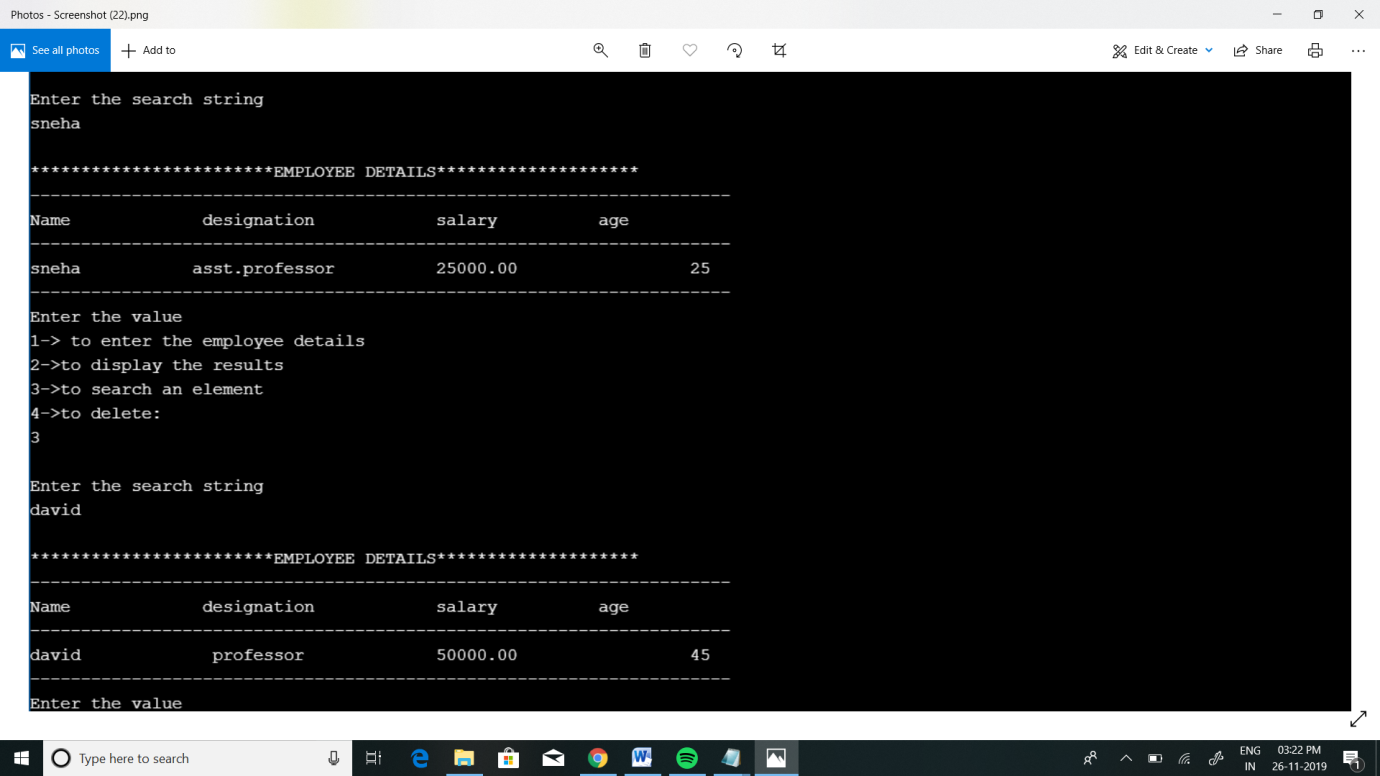
**Fig**.2.Insertion



**Fig**.3.Display



**Fig**.4.Deletion



**Fig**.5.Searching

**CONCLUSION**

By the end of this project, my team and I conclude that we took a lot of effort to complete this project since we used the concept of linked list to do every operation.

At last we thank each and every one who helped us to finish our project “EMPLOYEE MANAGEMENT USING LINKED LIST”.

At the end of this coursework I was able to understand the uses and advantages of the LINK LIST in everyday life.

This project is intended to ease the needs of schools,colleges as well as the companies in a employee management sector by embedding all the operations taking place in a schools, colleges etc.

The project can be enhanced in the future version by updating the employee records and giving a password to the user to login.

**FUTURE ENHANCEMENT**

As we end this stage of the project, we would like to recommend other researches in this field or related to :

1. Add visual effects to the program which will get more attention for the user when implementing an algorithm
2. Add sound effects to the program to make it more attractive.
3. Use a more powerful programming like visual C++ or VB .NET , which has more modules and facilities.
4. We can enhance this project by making it Employee Payroll Management system which is an application that enables users to create and store Employee Records. The application also provides facilities of a payroll system that enables the user to generate Payslips.

**REFERENCES**

1. [**www.codeproject.com**](http://www.codeproject.com)
2. [**https://www.codewithc.com**](https://www.codewithc.com)
3. **https://github.com**