VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

DATA STRUCTURES AND ALGORITHMS

REVIEW 3

TITLE:-WORKING SECTOR MANAGEMENT

FACULTY: PROF. JOSHVA DEVADAS T

SLOT: B1/20

PRESENTED BY:-

MANISH GUPTA CH	18BCE0455
SAI TEJA.K	18BCE0021
CHARAN L	18BCE0365

ACKNOWLEDGEMENT

We are extremely grateful to**Dr. SARAVANAN R,** Dean of the School of Computer Science And Engineering, VIT Vellore, for extending the facilities of the School towards our project and for his unstinting support.

We wish to express our sincere thanks and deep sense of gratitude to our project guide,**Dr. JOSHVA DEVADAS T,** Associate Professor, School of Computer Science and Engineering, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

Manish **Teja** **Charan**

TABLES OF CONTENT:

ABSTRACT:

The big MNC companies will be able to provide jobs opportunities to people in their respective field.At the same time we can add and update information of a worker.The company without any difficult can search for data of any worker without any difficult.

The authority of the company can use this software to monitor salaries of the workers,number of workers employed and number of days worked by each worker

PROBLEM DESCRIPTION:

In this project of ours we will see how we will store the information of the workers who are working in a particular sector. Here, you can create a new account, update information of an existing account, view and manage transactions, check the details of an existing account, remove existing account and view workers list.

This will make our job of searching any particular data about a worker very easy and efficient.This type of systematic organization will help any company to run smoothly and effectively without any problem.

This will help to keep a check on all the workers and will also help in maintaining a proper record and data of every bit of information.

PROJECT MODULE / TASKS (LISTING)

Our program can perform following functions:-

a)insert

b)delete

c)display

d)no.of workers

DESCRIPTION OF THE MODULES

Our program can perform following functions:-

a.Void list of employes

we can view the list of employees

b.Void employee details

This function can view the employee details

c.Void modify the employee

we can modify the employee details

d.void add employee details

we can insert employee details

e.void remove

we can delete the employee detail

ALGORITHM USED

1) First user will be asked to give the input.

2) This function can add and store the information of new and existing workers

3) Now start a loop to continue forward

4) In the insertion we can add

1. ID NUMBER
2. NAME OF THE PERSON
3. SALARY
4. MONTH OF JOINING
5. SECTOR

5) And then we will add another function delete. It will helpful us to delete the certain worker

6) And after we will display the workers list

ALL THE FUNCTIONS ARE STORED BY USING LINKED LIST

In this project we use file allocation system to save data

CODE:

```
#include<iostream>
#include<conio.h>
#include<cstdlib>
#include<cstring>
#include<cstdio>
#include<windows.h>
#include<unistd.h>
using namespace std;

class employee{
   private:
      // Variables for employee details
      char name[30];
      char id[5];
      char designation[10];
      int age;
      int ctc;
      int experience;

      // Utility functions
      void waitForEnter(void){
         cout<<"\n\n\n Press enter to go back \n\n";
         cin.get();
         cin.get();
      }
```

```
// Functions to perform desired actions
void listEmployees(void){ //To list total employees with Name, Id and Designation
   system("cls");
   FILE *file;
   file= fopen("data.txt", "r");
   cout<<"\n\t     List of Employees\n";
   cout<<"\n-----------------------------------------------";
   cout<<"\n NAME       |    ID    |    DESIGNATION\n";
   cout<<"-----------------------------------------------";
   while(fscanf(file, "%s %s %s %d %d %d", &name[0], &id[0] , &designation[0], &age, &ctc, &experience)!= EOF)
      cout<<"\n"<<name<<"\t\t"<<id<<"\t\t"<<designation;
   fclose(file);
   waitForEnter();
}

void showDetails(void){ //Displays all details according to Employee's id
   system("cls");
   FILE *file;
   char checkId[5];
   cout<<"\n\nEnter Employee ID: ";
   cin>>checkId;
   file= fopen("data.txt", "r");
   while(fscanf(file, "%s %s %s %d %d %d", &name[0], &id[0] , &designation[0], &age, &ctc, &experience)!=EOF)
      if(strcmp(checkId,id)==0){
            cout<<"\n---------------------";
         cout<<"\nName: "<<name;
```

```
            cout<<"\n----------------------";
            cout<<"\nId: "<<id;
            cout<<"\n----------------------";
            cout<<"\nDesignation: "<<designation;
            cout<<"\n----------------------";
            cout<<"\nAge: "<<age;
            cout<<"\n----------------------";
            cout<<"\nCTC: "<<ctc;
            cout<<"\n----------------------";
            cout<<"\nExperience: "<<experience;
            cout<<"\n----------------------";
        }
    fclose(file);
    waitForEnter();
}

void editExisting(void){ //edits Designation and CTC of an employee
    system("cls");
    char checkId[5];
    cout<<"\nEnter employee id: ";
    cin>>checkId;
    char newDesignation[10];
    cout<<"\n-----------------------------";
    cout<<"\nEnter new designation: ";
    cin>>newDesignation;
    int newCtc;
    cout<<"-----------------------------";
```

```
        cout<<"\nEnter new CTC: ";
        cin>>newCtc;
        FILE *file, *tempfile;
        file= fopen("data.txt", "r");
        tempfile= fopen("temp.txt", "w");
        while(fscanf(file, "%s %s %s %d %d %d", &name[0], &id[0] , &designation[0], &age, &ctc,
&experience)!=EOF){
            if(strcmp(checkId, id)==0)
                fprintf(tempfile, "%s %s %s %d %d %d \n", name, id, newDesignation, age, newCtc,
experience );
            else
                fprintf(tempfile, "%s %s %s %d %d %d \n", name, id, designation, age, ctc, experience );
        }
        fclose(file);
        fclose(tempfile);
        int isRemoved= remove("data.txt");
        int isRenamed= rename("temp.txt", "data.txt");
        waitForEnter();
    }

    void addNewEmployee(void){ //adding records
        system("cls");
        cout<<"\n----------------------------------------";
        cout<<"\n Enter First Name of Employee: ";
        cin>>name;
        cout<<"\n----------------------------------------";
        cout<<"\n Enter Employee ID [max 4 digits]: ";
        cin>>id;
```

```
cout<<"\n----------------------------------------";
cout<<"\n Enter Designation: ";
cin>>designation;
cout<<"\n----------------------------------------";
cout<<"\n Enter Employee Age: ";
cin>>age;
cout<<"\n----------------------------------------";
cout<<"\n Enter Employee CTC: ";
cin>>ctc;
cout<<"\n----------------------------------------";
cout<<"\n Enter Employee Experience: ";
cin>>experience;
cout<<"\n----------------------------------------";

char ch;
cout<<"\nEnter 'y' to save above information\n";
cin>>ch;
if(ch=='y'){
   FILE  *file;
   file= fopen("data.txt","a");
   fprintf(file, "%s %s %s %d %d %d \n", name, id, designation, age, ctc, experience );
   fclose(file);
   cout<<"\nNew Employee has been added to database\n";
}
else
   addNewEmployee();
waitForEnter();
```

```
}

void deleteEmployeeDetails(void){ //removing records
   system("cls");
   char checkId[5];
   cout<<"\n---------------------------------";
   cout<<"\nEnter Employee Id To Remove: ";
   cin>>checkId;
   char ch;
   cout<<"---------------------------------";
   cout<<"\n\n\n\n\nCONFIRMATION\nEnter 'y' To Confirm Deletion \n";
   cin>>ch;
   if(ch=='y'){
      FILE *file, *tempfile;
      file= fopen("data.txt", "r");
      tempfile= fopen("temp.txt", "w");
      while(fscanf(file, "%s %s %s %d %d %d", &name[0], &id[0] , &designation[0], &age, &ctc,
&experience)!=EOF)
         if(strcmp(checkId, id)!=0)
            fprintf(tempfile, "%s %s %s %d %d %d \n", name, id, designation, age, ctc, experience );
      fclose(file);
      fclose(tempfile);
      int isRemoved= remove("data.txt");
      int isRenamed= rename("temp.txt", "data.txt");
      cout<<"\nRemoved Successfully\n";
      waitForEnter();
   }
   else
```

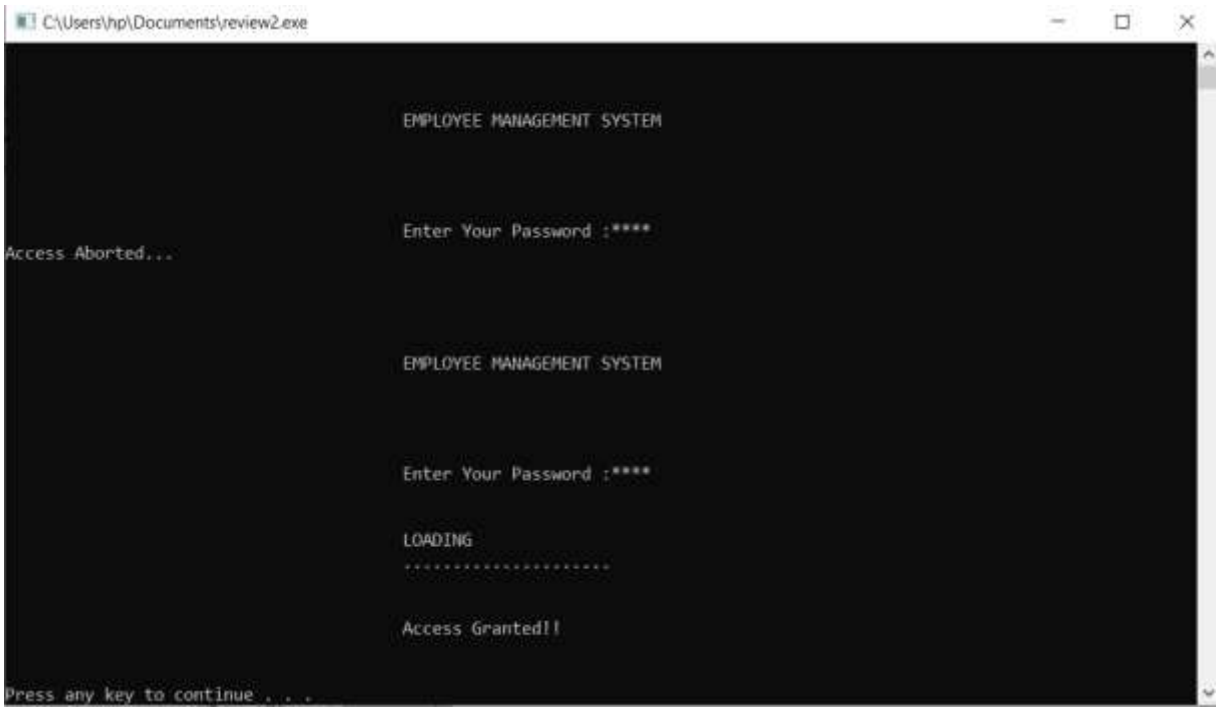
```
        deleteEmployeeDetails();
    }



public:
    // Function to serve as end point
    void options(void){ //menu
    int login(); //login declaration
        login();//login screen
       while(true){
        system("cls");


        // Options to choose an action
        cout<<"\n\t\t\t>>>>>>>>>  EMPLOYEE MANAGEMENT SYSTEM  <<<<<<<<<";
        cout<<"\n";
        cout<<"\n\t\t\t-----------------------------------------------";
        cout<<"\n\t\t\tENTER   1:   To View List of Employees";
        cout<<"\n\t\t\t-----------------------------------------------";
        cout<<"\n\t\t\tENTER   2:   To View Employee Details";
        cout<<"\n\t\t\t-----------------------------------------------";
        cout<<"\n\t\t\tENTER   3:   To Modify Existing Employee Details";
        cout<<"\n\t\t\t-----------------------------------------------";
        cout<<"\n\t\t\tENTER   4:   To Add New Employee Details";
        cout<<"\n\t\t\t-----------------------------------------------";
        cout<<"\n\t\t\tENTER   5:   To Remove an Employee Details";
        cout<<"\n\t\t\t-----------------------------------------------";
        cout<<"\n\t\t\tENTER   0:   To Exit     ";
```

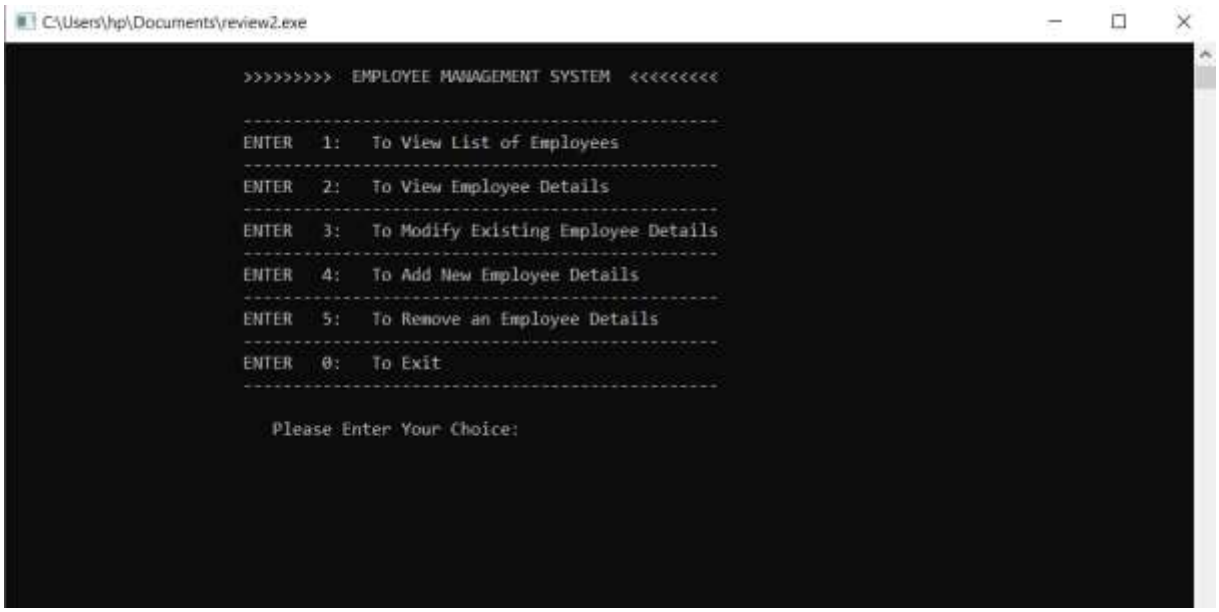
```
        cout<<"\n\t\t\t-----------------------------------------------";
        cout<<"\n\n\t\t\t   Please Enter Your Choice: ";

        // Taking the action input
        int choice;
        cin>>choice;
        // Calling relevant function as per choice
        switch (choice) {
           case 0:
                system("CLS");
              cout<<"\n\nEMPLOYEE MANAGEMENT SYSTEM \n\n Brought To You By code-
projects.org\n\n ";
              Sleep(10);
              return;
           case 1:
              listEmployees();
              break;
           case 2:
              showDetails();
              break;
           case 3:
              editExisting();
              break;
           case 4:
              addNewEmployee();
              break;
           case 5:
              deleteEmployeeDetails();
```

```
                break;
            default:
                cout<<"\n Sorry! I don't understand that! \n";
                break;
        }

    }
  }

};

int main(){
   // Call the options function
   employee e;
   e.options();
   return 0;
}

int login(){ //login procedure
  string pass ="";
  char ch;
  cout <<"\n\n\n\n\t\t\t\tEMPLOYEE MANAGEMENT SYSTEM";
  cout <<"\n\n\n\n\n\t\t\t\tEnter Your Password :";
  ch = _getch();
  while(ch != 13){//character 13 is enter
    pass.push_back(ch);
    cout << '*';
```

```
    ch = _getch();
  }
  if(pass == "pass"){
        cout<<"\n\n\n\t\t\t\t\tLOADING \n\t\t\t\t\t";
        for(int a=1;a<8;a++) // Change 'a<?' to how many * you want
        {
            Sleep(500);
            cout << "...";
        }
    cout << "\n\n\n\t\t\t\t\tAccess Granted!! \n\n\n";

    system("PAUSE");
    system("CLS");
  }else{
    cout << "\nAccess Aborted...\n";
    login();
  }
}
```

OUT PUT:


C:\Users\hp\Documents\review2.exe
EMPLOYEE MANAGEMENT SYSTEM
Enter Your Password :****
Access Aborted...
EMPLOYEE MANAGEMENT SYSTEM
Enter Your Password :****
LOADING
.....
Access Granted!!
Press any key to continue . . .


C:\Users\hp\Documents\review2.exe
>>>>>>>>> EMPLOYEE MANAGEMENT SYSTEM <<<<<<<<<
ENTER 1: To View List of Employees
ENTER 2: To View Employee Details
ENTER 3: To Modify Existing Employee Details
ENTER 4: To Add New Employee Details
ENTER 5: To Remove an Employee Details
ENTER 0: To Exit
Please Enter Your Choice:

****END***