

**KAKATIYA GOVERNMENT COLLEGE
HANAMKONDA**

STUDENTS' STUDY PROJECT

on

Salary Management System

Academic year – 2021-2022

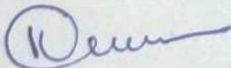


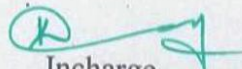
DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS

CERTIFICATE

This is to certify that the project report entitled "SALARY MANAGEMENT SYSTEM" submitted to the Department of Computer Science and Applications, Kakatiya Government College, Hanamkonda and it was carried out by the following students under my guidance.

1. Kontam Manasa - 006-20-2208
2. Kotha Saikumar - 006-20-2212
3. Lakavath Naresh - 006-20-2225
4. Nagelli Swathi – 006-20-2282
5. Moluguri Sindhu – 006-20-2265


Supervisor
(K.Ramesh)


Incharge


Principal

Contents

S/no	Title	Page no
1	Introduction	1
2	Objective	1
3	Existing system	1
4	Proposed system	2
5	Source Code	3
6	Output screens	24
7	Requirement	28
8	Advantages and disadvantages	28
9	Conclusion	28

Introduction

This project salary management system includes some facilities of managing the salary of the employees. It also save the records of employee's details and administrator can search, display, modification, delete etc. This software searches the employee data which is store in the record and after that print the salary report of the employee.

The software used for small company to maintain their records related to employees and salary

Objective

Salary management system will not responsible to calculate salary of employees working within particular organization but it will also able to perform other management activities which is required to perform their accounting operations for a particular organization such as maintain attendance of working employees, preparing pay slips and pay sheets for the employees and for other business transactions etc. This system will able to perform salary processing task and prepares monthly salary report for each employees. Accountant do not have to perform much activities in order to maintain ledger and preparing balance sheet as this system has been provided with features that can perform accounting operations with few steps such as automatic deductions of salary based of customer attendance, adding incentives and bonus to employees salary account, making tax deduction on some predefined rules. This system will also provide its admin to get details on list of employees who are on leave and their joining date. As each processing work will used different file system and file system will act as central database to maintain data integrity and reduce redundancy. Thus system has been embedded with two facilities under one common platform that means salary management system will able to acts as employee information system and salary management system.

Existing System

Previous salary management system works on manual system, so the system was not able to perform different activities at one time. Each department have to look after different activities, thus integrating information by collecting data from different department was not an easy task. Providing payroll solutions by the accounting department for the particular organization was not an easy task such as making analysis and generating reports on that basis, employee's reports and statutory reporting was time consuming process. Working employees were also not satisfied from their payment statement, as not providing clear figure for their working wages. The most important feature which was lacking in the previous system was file processing which may

sometimes results in disputes between working department and delay in rectifying the errors between working department of a particular organization.

Proposed System

Under this new system all working departments will be able to share information between each other and be able to work on same data thus eliminating the need of keeping records in different files, so enabling the features of data integrity and eliminating redundancy. Current system will eliminate the need of sending working sheets for final auditing to outside financial agencies in order to prepare final balance sheet. This system will easily be able to provide information on each working employees on various modules such as about their attendance and leave information, their extra earnings, deductions including taxes, loans etc. Even this new system using the concept of graphics will be able to generate graphical reports on the basis of financial reports and user query.

Source code:

```

//*****
//          HEADER FILE USED IN PROJECT
//*****

#include<conio.h>
#include<stdio.h>
#include<process.h>
#include<fstream.h>
#include<iomanip.h>
#include<string.h>
#include<ctype.h>
void middleadminmenu();
void printreportsalary(int empno);
void intromain();
int getsalary();
int getemployees();
void display_all();
void display_all_emp();
void salary_tabular();
void modify_recordsalary(int n);
void delete_recordsalary(int n);
void againopenandclose();
void againopenandclosecust();
int search(int p);
//void changeqty(int pr1,int q11);
//*****
//          CLASS USED IN PROJECT
//*****
//*****information*****
class employee
{
int emp_id;
char ename[25];
char address[35];
char phno[15];
char desg[35];
public:
void modifycust_data(int n1,char nm[15],char add[15],char q[15]);
void showempreport()
{
gotoxy(5,6);
cout<<"*****EMPLOYEE'S SALARY REPORT*****";
gotoxy(5,7);
cout<<"===== ";
gotoxy(5,8);
cout<<"EMP NO: ";
gotoxy(18,8);
cout<<emp_id;
gotoxy(25,8);
cout<<"NAME: ";
gotoxy(35,8);
cout<<ename;
gotoxy(50,8);
cout<<"ADDRESS: ";
gotoxy(65,8);
cout<<address;
gotoxy(5,9);
cout<<"PHONE NO.: ";
gotoxy(20,9);
cout<<phno;
gotoxy(35,9);
cout<<"DESIGNATION:";
gotoxy(48,9);
cout<<desg;
gotoxy(5,10);
cout<<"===== ";
}
}
int getempid()
{return emp_id;}
char *getempnm()

```

```

{ return ename;}
char *getempadd()
{ return address;}
char *getphno()
{ return phno;}
char *getdesg()
{ return desg;}
void show_cust()
{
gotoxy(5,7);
cout<<"=====";
gotoxy(10,8);
cout<<"EMP NO:";
gotoxy(25,8);
cout<<emp_id;
gotoxy(35,8);
cout<<"NAME OF EMP:";
gotoxy(54,8);
cout<<ename;
gotoxy(10,9);
cout<<"ADDRESS:";
gotoxy(25,9);
cout<<address;
gotoxy(10,10);
cout<<"PHONE NO.:";
gotoxy(25,10);
cout<<phno;
gotoxy(10,12);
cout<<"DESIGNATION:";
gotoxy(25,12);
cout<<desg;
gotoxy(5,14);
cout<<"=====";
}
void showallcust(int c)
{
gotoxy(1,c);
cout<<emp_id<<setw(17)<<ename<<setw(15)<<address<<setw(25)<<phno<<setw(15)<<desg;
}
void showempdatamulti()
{
gotoxy(5,7);
cout<<"=====";
gotoxy(10,8);
cout<<"EMP NO:";
gotoxy(30,8);
cout<<emp_id;
gotoxy(40,8);
cout<<"NAME OF EMP:";
gotoxy(60,8);
cout<<ename;
gotoxy(10,9);
cout<<"ADDRESS:";
gotoxy(30,9);
cout<<address;
gotoxy(10,10);
cout<<"PHONE NO.:";
gotoxy(30,10);
cout<<phno;
gotoxy(10,11);
cout<<"DESIGNATION:";
gotoxy(25,11);
cout<<desg;
gotoxy(5,11);
cout<<"=====";
}
void cust_input(int empid);
};
void employee::cust_input(int empid)
{ char ch;

```

```

gotoxy(5,7);
cout<<"=====";
gotoxy(10,8);
cout<<"EMP NO:";
emp_id=empid;
gotoxy(24,8);
cout<<emp_id;
gotoxy(35,8);
cout<<"NAME OF EMP:";
gotoxy(54,8);
cin>>ename;
gotoxy(10,9);
cout<<"ADDRESS:";
gotoxy(22,9);
cin>>address;
gotoxy(10,11);
cout<<"PHONE NO.:";
gotoxy(25,11);
cin>>phno;
gotoxy(10,12);
cout<<"DESIGNATION(M for manager,A for asst-manager,S for superwiser,W for worker:";
gotoxy(25,13);
cin>>ch;
switch(ch)
{
case 'M':strcpy(desg,"MANAGER");
break;
case 'A':strcpy(desg,"ASST-MANAGER");
break;
case 'S':strcpy(desg,"SUPERWISER");
break;
default:
strcpy(desg,"WORKER");
}
gotoxy(5,15);
cout<<"=====";
}
//*****modify employee details
void employee::modifycust_data(int n1,char nm[15],char add[15],char q[15])
{
char tmpnm[40],tmpnm2[40],tmpnm3[15];
gotoxy(5,14);
cout<<"=====WANT TO MODIFY =====";
gotoxy(10,15);
cout<<"EMP NO:";
emp_id=n1;
gotoxy(25,15);
cout<<emp_id;
gotoxy(40,15);
strcpy(ename,nm);
cout<<"NAME OF EMP:";
gotoxy(60,15);
cout<<ename;
gotoxy(10,17);
cout<<"Want to change the name of employee";
gotoxy(50,17);
int flag=0;
while(1)
{
gets(tmpnm);
if(strlen(tmpnm)!=0)
{
flag=1;
break;
}
if(strlen(tmpnm)==0)
{ flag=0;
break;
}
}
}

```



```

if(flag==1)
{ strcpy(ename,tmpnm);
}
gotoxy(1,18);
strcpy(address,add);
//***** NAME TO BE MODIFY
cout<<"CUSTOMER ADDRESS:";
gotoxy(20,18);
cout<<address;
gotoxy(45,18);
cout<<"Want to change the address";
gotoxy(10,19);
flag=0;
while(1)
{
    gets(tmpnm2);
    if(strlen(tmpnm2)!=0)
    {
        flag=1;
        break;
    }
    if(strlen(tmpnm2)==0)
    { flag=0;
      break;
    }
}
if(flag==1)
{ strcpy(address,tmpnm2);
}
//*****TO BE MODIFIED ENDS HERE
gotoxy(5,20);
strcpy(phno,q);
//***** phone no. TO BE MODIFY
cout<<"EMPLOYEE PHONE NO.:";
gotoxy(20,20);
cout<<phno;
gotoxy(45,20);
cout<<"Want to change the phone no.";
gotoxy(70,21);
flag=0;
while(1)
{
    gets(tmpnm3);
    if(strlen(tmpnm3)!=0)
    {
        flag=1;
        break;
    }
    if(strlen(tmpnm3)==0)
    { flag=0;
      break;
    }
}
if(flag==1)
{ strcpy(phno,tmpnm3);
}
char ch;
gotoxy(5,22);
cout<<"DESIGNATION(M for manager,A for asst-manager,S for superwiser,W for worker:";
gotoxy(25,23);
cin>>ch;
switch(ch)
{
case 'M':strcpy(desg,"MANAGER");
break;
case 'A':strcpy(desg,"ASST-MANAGER");
break;
case 'S':strcpy(desg,"SUPERWISER");
break;
default:

```

```

        strcpy(desg,"WORKER");
    }
    //*****MODIFIED ENDS HERE
    gotoxy(5,24);
    cout<<"===== ";
}
//*****class function employee modify ends here
//*****employee class ends here*****
//*****function to add the records in a file
void write_employee()
{
    ofstream objoff;
    employee cobj;
    objoff.open("employee.dat",ios::out|ios::app);
    clrscr();
    intomain();
    int r=getemployees();
    if(r>100)
    {
        r=1;
    }
    cobj.cust_input(r);
    objoff.write((char*)&cobj,sizeof(employee));
    objoff.close();
    gotoxy(10,20);
    cout<<"*****CUSTOMER RECORD SAVED***** ";
    cin.ignore();
    getch();
}
//*****function to add the record in a file ends here

//*****check the employee number already given or not*****
int getemployees()
{
    ifstream objjff;
    employee cust;
    int count=0;
    objjff.open("employee.dat",ios::binary);
    objjff.seekg(0,ios::beg);
    if(!objjff)
    {
        cout<<"File could not be open !! Press any Key...";
        getch();
    }
    //*****temporary hiding these lines
    while(objjff.read((char *) &cust, sizeof(employee)))
    {
        count++;
    }
    //*****jump to the last line
    objjff.seekg(count-sizeof(cust),ios::beg);
    objjff.read((char *) &cust, sizeof(employee));
    count=cust.getempid();
    count++;
    objjff.close();
    return count;
}
//*****check the employee number ends here
//*****
//      function to read specific record from file
//*****
void display_cust_sp(int n)
{
    ifstream objjfp;
    employee cust;
    int flag=0;
    objjfp.open("employee.dat",ios::binary);
    if(!objjfp)
    {
        cout<<"File could not be open !! Press any Key...";
    }
}

```

```

void deletecust_record(int n)
{
    employee cust;
    ifstream inFile;
    inFile.open("employee.dat",ios::binary);
    if(!inFile)
    {
        cout<<"File could not be open !! Press any Key...";
        getch();
        return;
    }
    int flag=0;
    while(inFile.read((char *) &cust, sizeof(employee)))
    {
        if(cust.gettempid()==n)
        { clrscr();
          intromain();
          cust.showempdatamulti();
          flag=1;
        }
    }
    inFile.close();
    char ch;
    if(flag==0)
        cout<<"\n\nrecord not exist";
    else
    {
        //*****deletion of the records starts from here
        gotoxy(1,15);
        cout<<"*****";
        gotoxy(5,16);
        cout<<"=====DO YOU WANT TO DELETE THE RECORDS GIVEN ABOVE[YES(Y) OR NO (N)=====";
        gotoxy(2,17);
        cin>>ch;
        if (toupper(ch)=='Y')
        {
            ofstream outFile;
            outFile.open("Temp2.dat",ios::binary);
            ifstream objjiff("employee.dat",ios::binary);
            objjiff.seekg(0,ios::beg);
            while(objjiff.read((char *) &cust, sizeof(employee)))
            {
                if(cust.gettempid()!=n)
                {
                    outFile.write((char *) &cust, sizeof(employee));
                }
            }
            outFile.close();
            objjiff.close();
            remove("employee.dat");
            rename("Temp2.dat", "employee.dat");
            againopenandclosecust();
            gotoxy(30,20);
            cout<<"-----Record Deleted-----";
        }
    }
    getch();
}
//*****delete record ends*****
void againopenandclosecust()
{
    ifstream inFile;
    employee cust;
    inFile.open("employee.dat",ios::binary);
    if(!inFile)
    {
        getch();
        return;
    }
    while(inFile.read((char *) &cust, sizeof(employee)))

```

```

        {
        }
        inFile.close();
    }
    //*****search the employee*****
    int searchcust(int p)
    {
        employee cust;
        int tmpprt=0;
        ifstream inFile;
        inFile.open("employee.dat",ios::binary);
        if(!inFile)
        {
            cout<<"File could not be open !! Press any Key...";
            getch();
            return -1;
        }
        int flag=0;
        while(inFile.read((char *) &cust, sizeof(employee)))
        {
            if(cust.getempid()==p)
            { clrscr();
              intomain();
              cust.showempdatamulti();
              flag=1;
              tmpprt=(int)inFile.tellg();
              break;
            }
        }
        inFile.close();
        if(flag==0)
            return 1;
            //cout<<"\n\nrecord not exist";
        else
        {
            return tmpprt;
        }
    }
    //*****employee information ends*****
    class salary
    {
        int salid;
        int eid;
        float basicsal;
        float hra,da,ta,pf;
        float netsal;
        public:
        salary()
        {
            basicsal=hra=da=ta=pf=netsal=0;
        }
        float getbasic()
        {
            return basicsal;
        }
        int getid()
        { return eid;}
        void modifydatasalary(int n1,int empid2,float bas,char dsg[15]);
    //*****
    void showdatamulti()
    {
        gotoxy(5,7);
        cout<<"=====";
        gotoxy(5,8);
        cout<<"SALARY NO:";
        gotoxy(30,8);
        cout<<salid;
        gotoxy(45,8);
        cout<<"EMP ID:";
        gotoxy(55,8);
        cout<<eid;
    }

```

```

gotoxy(5,9);
cout<<"BASIC SALARY:";
gotoxy(27,9);
cout<<basicsal;
gotoxy(35,9);
cout<<"HOUSE RENT ALLOWANCE: ";
gotoxy(60,9);
cout<<hra;
gotoxy(5,10);
cout<<"DEARNESS ALLOWANCE: ";
gotoxy(27,10);
cout<<da;
gotoxy(35,10);
cout<<"TRAVELLING ALLOWANCE:";
gotoxy(60,10);
cout<<ta;
gotoxy(5,11);
cout<<"PROVIDENT FUND:";
gotoxy(22,11);
cout<<pf;
gotoxy(35,11);
cout<<"NET SALARY:";
gotoxy(60,11);
cout<<netsal;
gotoxy(5,12);
cout<<"=====";
}
//*****
void create_sal(int m1,char choice[15],int e)
{
//*****
gotoxy(5,7);
cout<<"=====";
gotoxy(5,8);
cout<<"SALARY NO:";
salid=rn1;
gotoxy(30,8);
cout<<salid;
eid=e;
gotoxy(45,8);
cout<<"EMP ID:";
gotoxy(55,8);
cout<<eid;
gotoxy(5,9);
cout<<"BASIC SALARY:";
gotoxy(20,9);
cin>>basicsal;
//*****
gotoxy(15,10);
cout<<"*****MODIFY**HRA TA DA PF AND NET SALARY *****";
if(strcmp(choice,"MANAGER")==0)
{
hra=(basicsal*30)/100;
da=(basicsal*28)/100;
ta=(basicsal*15)/100;
pf=(basicsal*10)/100;
netsal=(basicsal+hra+da+ta)-pf;
}else if(strcmp(choice,"ASST-MANAGER")==0)
{
hra=(basicsal*28)/100;
da=(basicsal*25)/100;
ta=(basicsal*12)/100;
pf=(basicsal*9)/100;
netsal=(basicsal+hra+da+ta)-pf;
}else if(strcmp(choice,"SUPERVISOR")==0)
{
hra=(basicsal*25)/100;
da=(basicsal*22)/100;
ta=(basicsal*10)/100;
pf=(basicsal*8)/100;
}
}

```

```

        netsal=(basicSal+hra+da+ta)-pf;
    }else
    {
        hra=(basicSal*20)/100;
        da=(basicSal*18)/100;
        ta=(basicSal*8)/100;
        pf=(basicSal*7)/100;
        netsal=(basicSal+hra+da+ta)-pf;
    }
    gotoxy(5,11);
    cout<<"HOUSE RENT ALLOWANCE: ";
    gotoxy(28,11);
    cout<<hra;
    gotoxy(35,11);
    cout<<"DEARNESS ALLOWANCE: ";
    gotoxy(60,11);
    cout<<da;
    gotoxy(5,12);
    cout<<"TRAVELLING ALLOWANCE:";
    gotoxy(30,12);
    cout<<ta;
    gotoxy(45,12);
    cout<<"PROVIDENT FUND:";
    gotoxy(65,12);
    cout<<pf;
    gotoxy(5,13);
    cout<<"NET SALARY:";
    //netsal=(basicSal+hra+ta+da)-pf;
    gotoxy(30,13);
    cout<<netsal;
    gotoxy(5,15);
    cout<<"-----";
}
void show_salary()
{
    //-----
    gotoxy(5,7);
    cout<<"-----";
    gotoxy(5,8);
    cout<<"SALARY NO:";
    gotoxy(30,8);
    cout<<salid;
    gotoxy(45,8);
    cout<<"EMP ID:";
    gotoxy(55,8);
    cout<<eid;
    gotoxy(5,9);
    cout<<"-----";
    gotoxy(5,10);
    cout<<"BASIC SALARY:";
    gotoxy(30,10);
    cout<<basicSal;
    gotoxy(40,10);
    cout<<"HOUSE RENT ALLOWANCE: ";
    gotoxy(68,10);
    cout<<hra;
    gotoxy(5,11);
    cout<<"DEARNESS ALLOWANCE: ";
    gotoxy(30,11);
    cout<<da;
    gotoxy(40,11);
    cout<<"TRAVELLING ALLOWANCE:";
    gotoxy(68,11);
    cout<<ta;
    gotoxy(5,12);
    cout<<"PROVIDENT FUND:";
    gotoxy(30,12);
    cout<<pf;
    gotoxy(5,13);
    cout<<"-----";
}

```

```

gotoxy(5,14);
cout<<"NET SALARY:";
gotoxy(30,14);
cout<<netsal;
gotoxy(5,16);
cout<<"=====";
}
//-----show data tabular form-----
void showallsalary(int c)
{
    gotoxy(1,c);
    cout<<salid<<setw(10)<<eid<<setw(10)<<basicsal<<setw(10)<<hra;
    cout<<setw(10)<<da<<setw(10)<<ta<<setw(10)<<pf<<setw(12)<<netsal;
}

//-----ends here-----
int retsalid()
{return salid;}
}; //class ends here
//-----
// global declaration for stream object, object
//-----
//-----modify salary
void salary::modifydatasalary(int n1,int empid2,float bas,char dsg[15])
{
float tmpbsal=0;
gotoxy(5,14);
cout<<"=====WANT TO MODIFY =====";
gotoxy(10,15);
cout<<"SALARY NO:";
salid=n1;
gotoxy(25,15);
cout<<salid;
gotoxy(40,15);
cout<<"EMP ID:";
eid=empid2;
gotoxy(50,15);
cout<<eid;
gotoxy(10,16);
cout<<"BASIC SALARY";
gotoxy(30,16);
basicsal=bas;
cout<<basicsal;
gotoxy(10,17);
cout<<"Want to change the basic salary[0 for no]";
gotoxy(50,17);
cin>>tmpbsal;
if(tmpbsal!=0)
{
    basicsal=tmpbsal;
}
gotoxy(5,18);
cout<<"*****MODIFY**HRA TA DA PF AND NET SALARY *****";
if(strcmp(dsg,"MANAGER")==0)
{
    hra=(basicsal*30)/100;
    da=(basicsal*28)/100;
    ta=(basicsal*15)/100;
    pf=(basicsal*10)/100;
    netsal=(basicsal+hra+da+ta)-pf;
}else if(strcmp(dsg,"ASST-MANAGER")==0)
{
    hra=(basicsal*28)/100;
    da=(basicsal*25)/100;
    ta=(basicsal*12)/100;
    pf=(basicsal*9)/100;
    netsal=(basicsal+hra+da+ta)-pf;
}else if(strcmp(dsg,"SUPERVISOR")==0)
{
    hra=(basicsal*25)/100;

```

```

da=(basicsal*22)/100;
ta=(basicsal*10)/100;
pf=(basicsal*8)/100;
netsal=(basicsal+hra+da+ta)-pf;
}else
{
    hra=(basicsal*20)/100;
    da=(basicsal*18)/100;
    ta=(basicsal*8)/100;
    pf=(basicsal*7)/100;
    netsal=(basicsal+hra+da+ta)-pf;
}
gotoxy(5,20);
cout<<"HRA: "<<hra;
gotoxy(20,20);
cout<<"DA: "<<da;
gotoxy(35,20);
cout<<"TA: "<<ta;
gotoxy(55,20);
cout<<"PF: "<<pf;
gotoxy(5,21);
cout<<"-----";
gotoxy(25,22);
cout<<"NET SALARY: "<<netsal;
gotoxy(5,23);
cout<<"-----";
gotoxy(5,25);
cout<<"=====
}
//*****class function outside
fstream fp;
salary sal;
//-----
//    function to write in file
//-----
void write_salary()
{fstream objjff("employee.dat",ios::binary);
char str1[15];
employee eobj;
int flag=0,eid;
fp.open("salary.dat",ios::binary|ios::out|ios::app);
clrscr();
intromain();
int mn=getsalary();
if(rnn>100)
{
mn=1;
}
cout<<"Enter the employee no:";
cin>>eid;
while(objjff.read((char*)&eobj,sizeof(employee)))
{
if(eobj.gettempid()==eid)
{
clrscr();
intromain();
eobj.show_cust();
strcpy(str1,eobj.getdesg());
flag=1;
break;
}
}
objjff.close();

if(flag==0)
{ cout<<"\n\nrecord not exist";
getch();
}else
{
if(flag==1)

```



```

{ clrscr();
  intromain();
  //*****
  sal.create_sal(rmn,str1,eid);
  fp.write((char*)&sal,sizeof(salary));
  gotoxy(10,20);
  cout<<"*****SALARY RECORD SAVED***** ";
}

} cin.ignore();
  fp.close();
  getch();
}

//*****check the salary number already given or not*****
int getsalary()
{
  ifstream objiff;
  salary st;
  int count=0;
  objiff.open("salary.dat",ios::binary);
  objiff.seekg(0,ios::beg);
  if(!objiff)
  {
    cout<<"File could not be open !! Press any Key...";
    getch();
  }
  //*****temporary hiding these lines
  while(objiff.read((char *) &st, sizeof(salary)))
  {
    count++;
  }
  //*****jump to the last line
  objiff.seekg(count-sizeof(st),ios::beg);
  objiff.read((char *) &st, sizeof(salary));
  count=st.retsalid();
  count++;
  objiff.close();
  return count;
}
//*****
// function to read specific record from file
//*****
void display_sp_sal(int n)
{
  int flag=0;
  fp.open("salary.dat",ios::in);
  if(!fp)
  {
    cout<<"File could not be open !! Press any Key...";
    getch();
    return;
  }
  while(fp.read((char*)&sal,sizeof(salary)))
  {
    if(sal.retsalid()==n)
    {
      clrscr();
      intromain();
      sal.show_salary();
      flag=1;
    }
  }
  fp.close();
  if(flag==0)
  cout<<"\n\nrecord not exist";
  getch();
}

//*****
// function before display salary report check the employee id

```

```

//.....
int before_order()
{ int f=-1,num=0;
  employee cust;
  clrscr();
  intromain();
  gotoxy(10,8);
  cout<<"*****ENTER THE EMP ID TO BE SEARCHED*";
  gotoxy(55,9);
  cin>>num;
ifstream inFile;
  inFile.open("employee.dat",ios::binary);
  if(!inFile)
  {
    cout<<"File could not be open !! Press any Key...*";
    getch();
    return -1;
  }
  while(inFile.read((char *) &cust, sizeof(employee)))
  {
    if(cust.getempid()==num)
    { clrscr();
      intromain();
      cust.showempdatamulti();
      f=1;
      break;
    }
  }
  inFile.close();
return f;
}
//.....
// INTRODUCTION FUNCTION
//.....
void intro()
{
  gotoxy(4,2);
  cout<<"*****EMPLOYEES-MANAGEMENT*****";
  gotoxy(12,4);
  cout<<"*****";
  gotoxy(5,5);
  cout<<"*****S=Y=S=T=E=M*****";
  gotoxy(50,10);
  cout<<"PROJECT*";
  gotoxy(50,12);
  cout<<"MADE BY : TUSHAR*";
  gotoxy(10,14);
  cout<<"-----SCHOOL : DOON PUBLIC SCHOOL-----*";
  getch();
}
//.....
// ADMINISTRATOR MENU1 FUNCTION
//.....
void admin_menu1()
{
  clrscr();
  char ch2;
  int num;
  clrscr();
  intromain();
  gotoxy(20,6);
  cout<<"*****ADMIN MENU*****";
  gotoxy(22,7);
  cout<<"1.CREATE EMPLOYEE DETAILS*";
  gotoxy(22,8);
  cout<<"2.DISPLAY ALL EMPLOYEE'S DETAILS*";
  gotoxy(22,9);
  cout<<"3.SEARCH RECORD(QUERY) *";
  gotoxy(22,10);
}

```

```

cout<<"4.MODIFY EMPLOYEE RECORDS";
gotoxy(22,11);
cout<<"5.DELETE EMPLOYEE RECORDS";
gotoxy(22,12);
cout<<"6.BACK TO MAIN MENU";
gotoxy(18,13);
cout<<"Please Enter Your Choice (1-6) ";
gotoxy(55,13);
//*****
ch2=getche();
switch(ch2)
{
    case '1': clrscr();
              write_employee();
              break;
    case '2': cust_tabular();
              break;
    case '3':
              clrscr();
              intromain();
              gotoxy(10,8);
              cout<<"*****ENTER THE EMP ID TO BE SEARCHED:";
              gotoxy(55,9);
              cin>>num;
              display_cust_sp(num);
              break;
    case '4':
              clrscr();
              intromain();
              gotoxy(10,8);
              cout<<"*****ENTER THE EMP ID TO BE SEARCHED AND MODIFY:";
              gotoxy(55,9);
              cin>>num;
              modify_cust_record(num);
              break;
    case '5':
              clrscr();
              intromain();
              gotoxy(10,8);
              cout<<"*****ENTER THE EMP ID TO BE SEARCHED AND TO DELETE:";
              gotoxy(55,9);
              cin>>num;
              deletecust_record(num);
              break;
    case '6': break;
    default:cout<<"\a";admin_menu1();
}
}
//*****MENU 1 CUSTOMERS ENDS HERE*****
//*****
//      ADMINSTRATOR MENU2 FUNCTION
//*****
void admin_menu()
{
    clrscr();
    char ch2;
    int num;
    //*****
    clrscr();
    intromain();
    gotoxy(20,6);
    cout<<"=====ADMIN MENU=====";
    gotoxy(22,7);
    cout<<"1.CREATE SALARY DATA OF EMPLOYEE";
    gotoxy(22,8);
    cout<<"2.DISPLAY ALL SALARY REPORT'S AVAILABEL";
    gotoxy(22,9);
    cout<<"3.SEARCH AND DISPLAY SALARY RECORD(QUERY) ";
    gotoxy(22,10);
    cout<<"4.MODIFY SALARY INFORMATION";
}

```

```

cout<<"4.MODIFY EMPLOYEE RECORDS";
gotoxy(22,11);
cout<<"5.DELETE EMPLOYEE RECORDS";
gotoxy(22,12);
cout<<"6.BACK TO MAIN MENU";
gotoxy(18,13);
cout<<"Please Enter Your Choice (1-6) ";
gotoxy(55,13);
//*****
ch2=getche();
switch(ch2)
{
    case '1': clrscr();
                write_employee();
                break;
    case '2': cust_tabular();
                break;
    case '3':
                clrscr();
                intromain();
                gotoxy(10,8);
                cout<<"*****ENTER THE EMP ID TO BE SEARCHED:";
                gotoxy(55,9);
                cin>>num;
                display_cust_sp(num);
                break;
    case '4':
                clrscr();
                intromain();
                gotoxy(10,8);
                cout<<"*****ENTER THE EMP ID TO BE SEARCHED AND MODIFY:";
                gotoxy(55,9);
                cin>>num;
                modify_cust_record(num);
                break;
    case '5':
                clrscr();
                intromain();
                gotoxy(10,8);
                cout<<"*****ENTER THE EMP ID TO BE SEARCHED AND TO DELETE:";
                gotoxy(55,9);
                cin>>num;
                deletecust_record(num);
                break;
    case '6': break;
    default:cout<<"\a";admin_menu1();
}
}
//*****MENU 1 CUSTOMERS ENDS HERE*****
//*****
//      ADMINSTRATOR MENU2 FUNCTION
//*****
void admin_menu()
{
    clrscr();
    char ch2;
    int num;
    //*****
    clrscr();
    intromain();
    gotoxy(20,6);
    cout<<"=====ADMIN MENU=====";
    gotoxy(22,7);
    cout<<"1.CREATE SALARY DATA OF EMPLOYEE";
    gotoxy(22,8);
    cout<<"2.DISPLAY ALL SALARY REPORT'S AVAILABEL";
    gotoxy(22,9);
    cout<<"3.SEARCH AND DISPLAY SALARY RECORD(QUERY) ";
    gotoxy(22,10);
    cout<<"4.MODIFY SALARY INFORMATION";
}

```

```

gotoxy(22,11);
cout<<"5.DELETE SALARY INFORMATION";
gotoxy(22,12);
cout<<"6.BACK TO MAIN MENU";
gotoxy(18,13);
cout<<"Please Enter Your Choice (1-6) ";
gotoxy(55,13);
//*****
ch2=getche();
switch(ch2)
{
    case '1': clrscr();
    write_salary();
    break;
    case '2': salary_tabular();
    break;
    case '3':
    //*****
    clrscr();
    intromain();
    gotoxy(10,8);
    cout<<"*****ENTER THE SALARY ID TO BE SEARCHED:";
    gotoxy(55,9);
    cin>>num;
    display_sp_sal(num);
    break;
    case '4':
    clrscr();
    intromain();
    gotoxy(10,8);
    cout<<"*****ENTER THE SALARY ID TO BE SEARCHED AND MODIFY:";
    gotoxy(55,9);
    cin>>num;
    modify_recordsalary(num);
    break;
    case '5':
    clrscr();
    intromain();
    gotoxy(10,8);
    cout<<"*****ENTER THE SALARY ID TO BE SEARCHED AND TO DELETE:";
    gotoxy(55,9);
    cin>>num;
    delete_recordsalary(num);
    break;
    case '6': break;
    default:cout<<"\a";admin_menu();
}
}
//*****
// THE MAIN FUNCTION OF PROGRAM
//*****
void main()
{
    char ch; int n=0;
    intro();
    do {
        //*****TEMPORARY*****
        clrscr();
        intromain(); gotoxy(20,6);
        cout<<"=====MAIN MENU=====";
        gotoxy(28,7); cout<<"01. SALARY REPORT GENERATOR";
        gotoxy(28,8); cout<<"02. ADMINISTRATOR";
        gotoxy(28,9); cout<<"03. EXIT";
        gotoxy(20,10); cout<<"=====";
        gotoxy(25,12); cout<<"Please Select Your Option (1-3) ";
        gotoxy(30,14);
        ch=getche();

        switch(ch)
        {
            case '1':

```

```

        clrscr();
        intromain();
        gotoxy(10,8);cout<<"*****ENTER THE EMP ID TO BE SEARCHED:";
        gotoxy(55,9);cin>>n;
        printreportsalary(n);
        getch();
        break;
        case '2': middleadminmenu();
            break;
        case '3':exit(0);
        default :cout<<"\a";
    }
    }while(ch!='3');
}

//*****main intro
void intromain()
{ clrscr();
  gotoxy(1,2);
  cout<<"*****EMPLOYEE*****MANAGEMENT****SYSTEM*****PROJECT*****";
  gotoxy(1,3);
  cout<<"*****";
}

//*****
//          DISPLAY ALL THE PRODUCT TABULAR FORM
//*****
void salary_tabular()
{
  int r=0,col=10;
  salary st;
  ifstream inFile;
  inFile.open("salary.dat",ios::binary);
  if(!inFile)
  {
    cout<<"File could not be open !! Press any Key...";

    getch();
    return;
  }
  display_all();
  while(inFile.read((char *) &st, sizeof(salary)))
  {
    if(r<=12)
    {
      r++;
      st.showallsalary(col);
      col++;
    }else
    {
      gotoxy(20,30); cout<<"-----press any key-----";
      getch(); clrscr();
      display_all();
      col=10;
      r=0;
    }
  }
  inFile.close();
  getch();
}

//*****tabular forms ends*****
//*****tabulars forms headings*****
//function to display all the records of salary
//*****
void display_all()
{
  clrscr();
  intromain();
  gotoxy(1,5);
  cout<<"*****EMPLOYEE'S SALARY'S DETAILS*****";
  gotoxy(1,6);
}

```

```

cout<<"=====";
gotoxy(1,7);
cout<<"SALNO"<<setw(10)<<"EMP NO"<<setw(10)<<"BASICSAL"<<setw(10)<<"HRA"<<setw(10)<<"DA"<<setw(10)<<" TA
"<<setw(10)<<" PF "<<setw(12)<<"NET SAL";
gotoxy(1,8);
cout<<"=====";
}
//*****
//          MODIFY RECORD
//*****
void modify_recordsalary(int n)
{
    salary st,temp;

    ifstream inFile;
    int fpos=-1;
    inFile.open("salary.dat",ios::binary);
    if(!inFile)
    {
        cout<<"File could not be open !! Press any Key...";
        getch();
        return;
    }
    int flag=0;
    while(inFile.read((char *) &st, sizeof(salary)))
    {
        if(st.retsalid()==n)
        {
            clrscr();
            intromain();
            st.showdatamulti();
            flag=1;
        }
    }
    inFile.close();
    if(flag==0)
        cout<<"\n\nrecord not exist";
    else
    {
        //*****modifying the records starts here
        fstream File;
        File.open("salary.dat",ios::binary|ios::in|ios::out);
        if(!File)
        {
            cout<<"File could not be open !! Press any Key...";
            getch();
            return;
        }
        while(File.read((char *) &st, sizeof(salary)))
        {
            if(st.retsalid()==n)
            {
                fpos=(int)File.tellg();
                break;
            }
        }
        File.seekp(fpos-sizeof(salary),ios::beg);
        ifstream objemp("employee.dat",ios::binary);
        employee ee;
        float basic=st.getbasic();
        int empid;
        empid=st.getid();
        //int fpos2=0;
        char ds[35];
        //*****search the employee record *****
        while(objemp)
        {
            objemp.read((char*)&ee,sizeof(ee));
            if(ee.getempid()==empid)
            {
                //fpos2=(int)File.tellg();
                break;
            }
        }
    }
}

```

```

}
//objemp.seekg(fpos2-sizeof(employee),ios::beg);
strcpy(ds,ee.getdesg());
objemp.close();
//*****
gotoxy(1,12);
cout<<"-----";
gotoxy(1,13);
cout<<"=====ENTER NEW VALUES FOR THE RECORDS GIVEN ABOVE=====";
temp.modifydatasalary(n,empid,basic,ds);
File.write((char *) &temp, sizeof(salary));
File.close();
}
}
//*****
// DELETED THE RECORD OF THE PRODUCTC NOT AVAILABLE
//*****
void delete_recordsalary(int n)
{
    salary st;
    ifstream inFile;
    inFile.open("salary.dat",ios::binary);
    if(!inFile)
    {
        cout<<"File could not be open !! Press any Key...";
        getch();
        return;
    }
    int flag=0;
    while(inFile.read((char *) &st, sizeof(salary)))
    {
        if(st.retsalid()==n)
        {
            clrscr();
            intomain();
            st.showdatamulti();
            flag=1;
        }
    }
    inFile.close();
    char ch;
    if(flag==0)
        cout<<"\n\nrecord not exist";
    else
    {
        //*****deletion of the records starts from here
        gotoxy(1,15);
        cout<<"-----";
        gotoxy(5,16);
        cout<<"=====DO YOU WANT TO DELETE THE RECORDS GIVEN ABOVE[YES(Y) OR NO (N)]=====";
        gotoxy(2,17);
        cin>>ch;
        if (toupper(ch)=='Y')
        {
            ofstream outFile;
            outFile.open("Temp1.dat",ios::binary);
            ifstream objjff("salary.dat",ios::binary);
            objjff.seekg(0,ios::beg);
            while(objjff.read((char *) &st, sizeof(salary)))
            {
                if(st.retsalid()!=n)
                {
                    outFile.write((char *) &st, sizeof(salary));
                }
            }
            outFile.close();
            objjff.close();
            remove("salary.dat");
            rename("Temp1.dat","salary.dat");
            againopenandclose();
            gotoxy(30,20);
            cout<<"-----Record Deleted-----";
            getch();
        }
    }
}
//*****delete record ends*****
void againopenandclose()
{

```



```

ifstream inFile;
salary st;
inFile.open("salary.dat",ios::binary);
if(!inFile)
{
    getch();
    return;
}
while(inFile.read((char *) &st, sizeof(salary)))
{}
inFile.close();
}
//*****search the salary*****
int search(int p)
{
    salary st;
    int tmp=0;
    ifstream inFile;
    inFile.open("salary.dat",ios::binary);
    if(!inFile)
    {
        cout<<"File could not be open !! Press any Key...";
        getch();
        return -1;
    }
    int flag=0;
    while(inFile.read((char *) &st, sizeof(salary)))
    {
        if(st.retsalid()==p)
        {
            clrscr();
            intomain();
            st.showdatamulti();
            flag=1;
            tmp=(int)inFile.tellg();
        }
    }
    break;
    inFile.close();
    if(flag==0)
        return 1;
    //cout<<"\n\nrecord not exist";
    else
        return tmp;
}
void middleadminmenu()
{
    char ch;
    do
    {
        //*****TEMPORARY*****
        clrscr();
        intomain();
        gotoxy(20,6);
        cout<<"=====EMPLOYEE'S-SALARY'S MENU=====";
        gotoxy(28,7); cout<<"01. EMPLOYEE'S MENU";
        gotoxy(28,8); cout<<"02. SALARY'S MENU";
        gotoxy(28,9); cout<<"03. BACK TO MAIN";
        gotoxy(20,10);
        cout<<"=====";
        gotoxy(25,12); cout<<"Please Select Your Option (1-3) ";
        gotoxy(30,14);
        ch=getche();
        switch(ch)
        {
            case '1': admin_menu1();
                    break;
            case '2': admin_menu();
                    break;
            case '3': break;
            default :cout<<"\a";admin_menu();
        }
    }while(ch!='3');
}

```

```

//*****print the report of salary of employees
void printreportsalary(int empno)
{
    ifstream objjff("employee.dat",ios::binary);
    employee empobj;
    int flag=0;
    if(!objjff)
    {
        cout<<"File could not be open !! Press any Key...";
        getch();
        return;
    }
    clrscr();
    while(objjff.read((char *) &empobj, sizeof(employee)))
    {
        if(empobj.getempid()==empno)
        { clrscr();
          intomain();
          empobj.showempreport();
          flag=1;
          break;
        }
    }
    int id=empobj.getempid();
    objjff.close();
    if(flag==0)
        cout<<"\n\nrecord not exist";
    else
    {
        ifstream objjsal("salary.dat",ios::binary);
        salary salobj;
        gotoxy(1,11);
        cout<<"SALNO"<<setw(10)<<"EMP NO"<<setw(10)<<"BASICSAL"<<setw(10)<<"HRA"<<setw(10)<<"DA"<<setw(10)<<"TA"
        "<<setw(10)<<" PF "<<setw(12)<<"NET SAL";
        gotoxy(1,12);
        cout<<"=====";
        int cno=13;
        int r=0;
        if(!objjsal)
        {
            cout<<"File could not be open !! Press any Key...";

            getch();
            return;
        }
        while(objjsal.read((char *) &salobj, sizeof(salary)))
        {
            if(salobj.getid()==id)
            {
                salobj.showallsalary(cno);
                cno++;

                if(r<=15)
                {
                    r++;
                }else
                {
                    gotoxy(20,30);      cout<<"-----press any key-----";
                    getch();      clrscr();      display_all();
                    cno=10; r=0;
                }
            }
        }
        objjsal.close();
    }
}
//*****
//                          END OF PROJECT
//*****

```

24

```
-----MAIN MENU-----  
01. SALARY REPORT GENERATOR  
02. ADMINISTRATOR  
03. EXIT  
-----
```

Please Select Your Option (1-3)

EMP NO	NAME	GRADE	DATE	STATUS
1	JOHN	100000	10/10/78	REGULAR
2	JANE	100000	10/10/78	REGULAR
3	JOHN	100000	10/10/78	REGULAR
4	JANE	100000	10/10/78	REGULAR

EMP NO	NAME	GRADE	DATE	STATUS
1	JOHN	100000	10/10/78	REGULAR
2	JANE	100000	10/10/78	REGULAR
3	JOHN	100000	10/10/78	REGULAR
4	JANE	100000	10/10/78	REGULAR

if choice is: 2

```

-----EMPLOYEE'S-SALARY'S MENU-----
01. EMPLOYEE'S MENU
02. SALARY'S MENU
03. BACK TO MAIN
-----
    
```

Please Select Your Option (1-3)

if choice is: 1

```

-----ADMIN MENU-----
1.CREATE EMPLOYEE DETAILS
2.DISPLAY ALL EMPLOYEE'S DETAILS
3.SEARCH RECORD(QUERY)
4.MODIFY EMPLOYEE RECORDS
5.DELETE EMPLOYEE RECORDS
6.BACK TO MAIN MENU
Please Enter Your Choice (1-6)
    
```

```

-----
EMP NO:      2      NAME OF EMP:    abc
ADDRESS:     nainital

PHONE NO.:   45787878
DESIGNATION(M for manager,A for asst-manager,S for superwiser,V for wor
ker:        M
-----
    
```

*****CUSTOMER RECORD SAVED*****

CUST.NO	NAME	ADDRESS	PHONE NO	DESIGNATION
1	aakash	haldwani	8596857423	ASST-MANAGER
2	abc	nainital	4578787878	MANAGER_

*****ENTER THE EMP ID TO BE SEARCHED:

1

```

-----
EMP NO:      1      NAME OF EMP:    aakash
ADDRESS:     haldwani
PHONE NO.:   8596857423
DESIGNATION: ASST-MANAGER
-----
    
```

*****ENTER THE EMP ID TO BE SEARCHED AND MODIFY:

1

```

=====
EMP NO:          1      NAME OF EMP:    aakash
ADDRESS:        haldwani
PHONE NO.:      8596857423
=====
*****ENTER NEW VALUES FOR THE RECORDS GIVEN ABOVE*****
*****WANT TO MODIFY *****
EMP NO:          1      NAME OF EMP:    aakash

  Want to change the name of employee
CUSTOMER ADDRESS: haldwani          Want to change the address

EMPLOYEE PHONE 8596857423          Want to change the phone no.
                                     8989898989
DESIGNATION(N for manager,A for asst-manager,S for superwiser,W for worker:
      A_
*****ENTER THE EMP ID TO BE SEARCHED AND TO DELETE:
                                     1

```

```

=====
EMP NO:          1      NAME OF EMP:    aakash
ADDRESS:        haldwani
PHONE NO.:      8989898989
=====

```

```

*****DO YOU WANT TO DELETE THE RECORDS GIVEN ABOVE[YES(Y) OR NO (N)]*****
y

```

-----Record Deleted-----

if choice is 2: salary details

```

-----ADMIN MENU-----
1.CREATE SALARY DATA OF EMPLOYEE
2.DISPLAY ALL SALARY REPORT'S AVAILABEL
3.SEARCH AND DISPLAY SALARY RECORD(QUERY)
4.MODIFY SALARY INFORMATION
5.DELETE SALARY INFORMATION
6.BACK TO MAIN MENU
Please Enter Your Choice (1-6)  _

```

ter the employee no:1

```

=====
SALARY NO:      3      EMP ID:    2
BASIC SALARY:  8900
*****MODIFY**HRA TA DA PF AND NET SALARY *****
HOUSE RENT ALLOWANCE: 2670  DEARNESS ALLOWANCE:  2492
TRAVELLING ALLOWANCE: 1335  PROVIDENT FUND:    890
NET SALARY:      14507
=====

```

*****SALARY RECORD SAVED***** _

```

=====
SALARY NO:          1          EMP ID:  1
-----
BASIC SALARY:      5500      HOUSE RENT ALLOWANCE:  1540
DEARNESS ALLOWANCE: 1375      TRAVELLING ALLOWANCE:  660
PROVIDENT FUND:    495
-----
NET SALARY:        8580
=====

```

*****ENTER THE SALARY ID TO BE SEARCHED AND MODIFY:
1

```

=====
SALARY NO:          1          EMP ID:  1
BASIC SALARY:      5500      HOUSE RENT ALLOWANCE:  1540
DEARNESS ALLOWANCE: 1375      TRAVELLING ALLOWANCE:  660
PROVIDENT FUND:    495      NET SALARY:  8580
=====

```

*****ENTER NEW VALUES FOR THE RECORDS GIVEN ABOVE*****
 *****WANT TO MODIFY*****
 SALARY NO: 1 EMP ID: 1
 BASIC SALARY 5500
 Want to change the basic salary[0 for no]

*****ENTER THE EMP ID TO BE SEARCHED:
2

*****EMPLOYEE'S SALARY REPORT*****
 =====
 EMP NO: 2 NAME : abc ADDRESS: nainital
 PHONE NO.: 4578787878 DESIGNATION: MANAGER
 =====

SALNO	EMP NO	BASICSAL	HRA	DA	TA	PF	NET SAL
3	2	8900	2670	2492	1335	890	14507_

Requirements

- Hardware required
 - Printer, to print the required documents of the project
 - Compact drive
 - RAM 512 MB or more then 512 MB
 - Harddisk: 80 GB or More then 80 GB
- Software required
 - Operating system : Windows XP
 - Turbo C++, for execution of program and MS-Word, for presentation of output.

Advantages

- An employee does not need to worry about their salary details
- This software reduces paper work
- It is easy to handle Employee's record
- This software saves the time
- Information of the employees stores permanently
- Employee can also get the details of their salary yearly or monthly.

Disadvantages

- This system suitable for only small organization.
- Online facility is not available.

Conclusion

This software is efficient in maintaining employee's details and can easily perform operations on salary of the employees. This software also reduces the workload of the department which works on maintaining the records of the employees.

In future, this system can launch web site for easy online data entry.