Educate Elevate Enlighten

JSS Mahavidyapeetha
JSS Science And Technology University
(Established Under JSS Science and Technology University Act No. 43 of 2013)
(Formerly Known as SJCE)

JSS
SCIENCE AND
TECHNOLOGY
UNIVERSITY
MYSURU

Title: Payroll Management System

Subject name: Data Structures and Algorithm

Subject Code: EC560

Event – 4

Submitted by

5 Semester ‘B’ Section

Roll No.	Name	USN No.
06	Akhilesh R	01JST18EC007
09	Arjun V Srivatsa	01JST18EC012
14	Bijwal B Jain	01JST18EC020
27	Mahendhar H N	01JST18EC052

Submitted to

Prof. Pavithra D R

Assistant Professor

Department of E & C

JSS S&TU, Mysuru

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

JSS SCIENCE AND TECHONOLOGY UNIVERSITY

MYSURU-570006

2020-2021

Abstract:

Payroll Management system is the system by which an employer can manage the salaries of the employees. It indicates salary, allowances, deductions and net payable to the employees. It also deals with the generation of payslips. Complicated computations that are performed manually can be automated using the payroll system. An payroll management software is beneficial for any kind of business enterprises. Human resource department performs various activities such as payroll processing, maintain salary details of every employee in the organisations, keeping track of deductions, loans, allowances and bonuses. Human resource department collectively works together to calculate the available leaves, encashment of leaves and loss of pay for all the employees and then process their salaries.

Problem Statement:

- Every company will have different departments like marketing, finance, information technology, research and development, human resources. Human Resources department is the crucial part of any organisation because employees are considered as the real assets and also the HR department needs to look after the employee welfare and make the facilities easier.
- With the increase in competition among business enterprises. Human Resources are considered to be most important part of any company. It is said that if any company has a human resource then it will definitely have many advantages. This is the reason why the importance of HR department is increasing day-by-day. With the continuous advancements in the latest technologies, the management of employees and their records have become tough to handle. To prevent this it is necessary to have payroll management system.

Contents

1. Introduction:

Basic operations users can perform via this program project that are based on file handling are adding new employee record, modifying employee record and deleting record, displaying one or all employee’s record. Besides these, payroll management also allows users to print the salary slip for a particular employee. This project gives total outcome of payroll management.

- **Addition of New Employee:**

This feature is under the public functions of class employee. The information handled in this feature are name, employee code, resignation, experience, age and salary.

- **Edit Employee Record:**

This function enables user to edit the employee added before. It can edit all the features mentioned in addition of new employee. Later, The pay slip is changed according to the edited value.

- **Delete Employee Record:**

Deletion is done of an employee record from Payroll management system project by entering the employee code

- **Print Employee Salary Slip:**

This feature enables the user to get the print of pay slip of an employer. The program Asks the employer code to print the pay slip.

- **Display Employee Record:**

Providing the employee code number, users can access all the provided information related to a particular employee via this function.

- **Display List of Employees:**

This feature displays the record of all employees added in file. The records are displayed in a tabular pattern containing information such as code name of the employee, phone number, date of joining, designation, grade and salary.

2.Theory:

- **C++**

C++ is a high-level language that has developed in mid of 1970's. It is built or derived from C language. It has object-oriented features which allows programmer to create objects within the code. This creation of objects makes programming easier, efficient in terms of time and space. C++ offers power and flexibility over the language.

C++ is a general-purpose programming language and widely used now a days for competitive programming. It has imperative, object-oriented and generic programming features. C++ runs on lots of platform like Windows, Linux, Unix, Mac etc.

- **C++ Classes and Objects**

Class:

The building block of C++ that leads to Object Oriented programming is a Class. It is a user defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A class is like a blueprint for an object.

When you define a class, you define a blueprint for a data type. This doesn't actually define any data, but it does define what the class name means, that is, what an object of the class will consist of and what operations can be performed on such an object.

A class definition starts with the keyword class followed by the class name; and the class body, enclosed by a pair of curly braces. A class definition must be followed either by a semicolon or a list of declarations.

The keyword **public** determines the access attributes of the members of the class that follows it. A public member can be accessed from outside the class anywhere within the scope of the class object. You can also specify the members of a class as **private** or **protected** which we will discuss in a sub-section.

Object:

Object is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated. A class provides the blueprints for objects, so basically an object is created from a class. We declare objects of a class with exactly the same sort of declaration that we declare variables of basic types.

- **C++ Functions:**

A function is a group of statements that together perform a task. Every C++ program has at least one function, which is **main()**, and all the most trivial programs can define additional functions.

You can divide up your code into separate functions. How you divide up your code among different functions is up to you, but logically the division usually is such that each function performs a specific task.

A function **declaration** tells the compiler about a function's name, return type, and parameters. A function **definition** provides the actual body of the function. The C++ standard library provides numerous built-in functions that your program can call.

A C++ function definition consists of a function header and a function body. Here are all the parts of a function −

- **Return Type** − A function may return a value. The **return_type** is the data type of the value the function returns. Some functions perform the desired operations without returning a value. In this case, the return_type is the keyword **void**.
- **Function Name** − This is the actual name of the function. The function name and the parameter list together constitute the function signature.
- **Parameters** − A parameter is like a placeholder. When a function is invoked, you pass a value to the parameter. This value is referred to as actual parameter or argument. The parameter list refers to the type, order, and number of the parameters of a function. Parameters are optional; that is, a function may contain no parameters.
- **Function Body** − The function body contains a collection of statements that define what the function does.

- **File Handling In C++:**

Opening a File:

A file must be opened before you can read from it or write to it. Either **ofstream**or **fstream** object may be used to open a file for writing. And ifstream object is used to open a file for reading purpose only.

Following is the standard syntax for open() function, which is a member of fstream, ifstream, and ofstream objects.

Closing a File:

When a C++ program terminates it automatically flushes all the streams, release all the allocated memory and close all the opened files. But it is always a good practice that a programmer should close all the opened files before program termination.

Writing to a File:

While doing C++ programming, you write information to a file from your program using the stream insertion operator (<<) just as you use that operator to output information to the screen. The only difference is that you use an **ofstream** or **fstream** object instead of the **cout** object.

Reading from a File:

You read information from a file into your program using the stream extraction operator (>>) just as you use that operator to input information from the keyboard. The only difference is that you use an **ifstream** or **fstream** object instead of the **cin** object.

3.Flowchart:

This figure below is the flow of architecture as the user end.

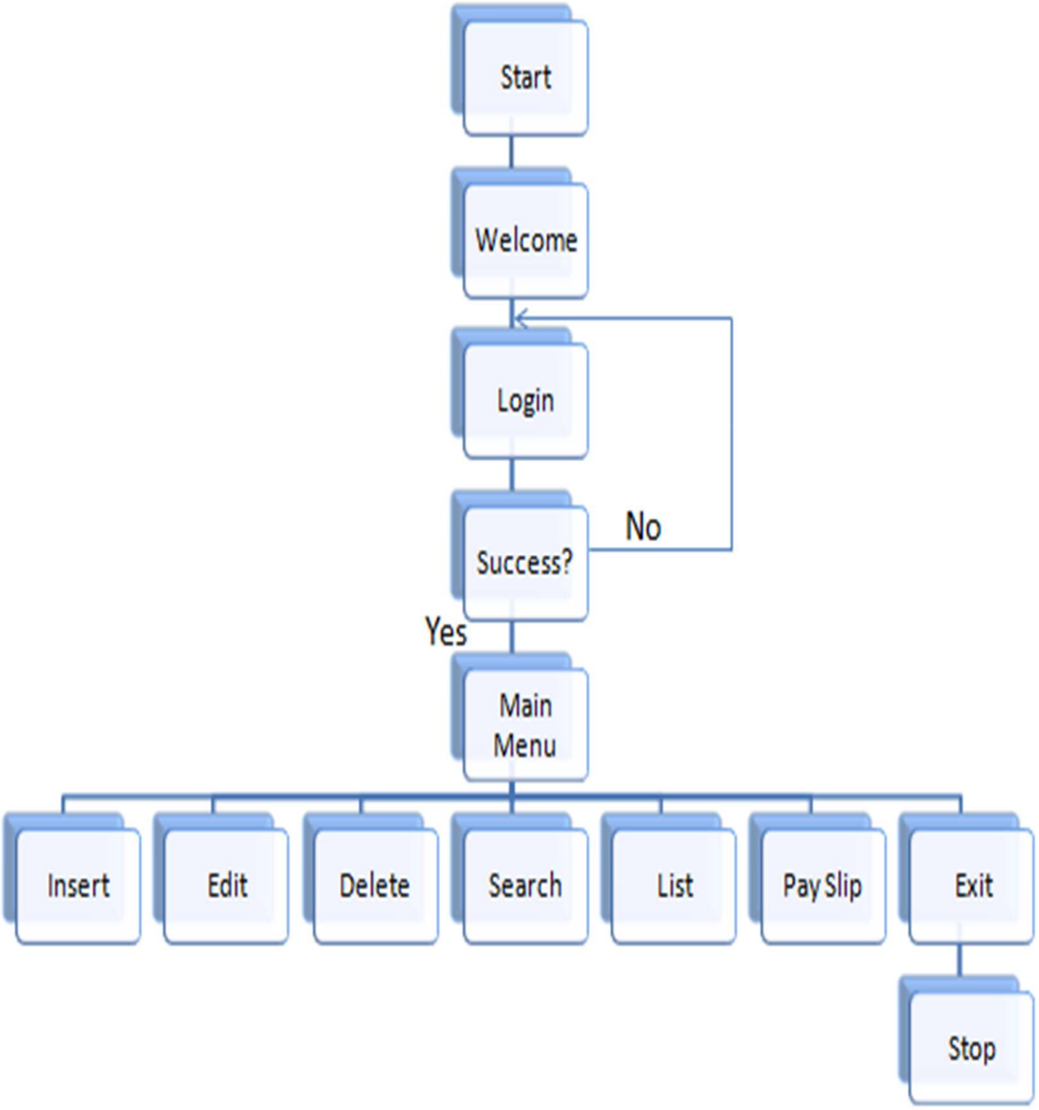


Figure 3.1: Flowchart of architecture

4.Working procedure:

Step 1:

Main function is executed first and the control to the followed by code. The function ‘intro()’ got executed and the welcome message is displayed and when the user proceed to the next step by pressing any key.

Step 2:

User prompted with a login screen here and a user with valid credentials can have the access to the software.

Step 3:

After the user with the valid credentials logged in, the data of previous employee records are retrieved.

Step 4:

After the user with the valid credentials logged in and successful retrieval, the user land on the home screen (the Main menu), and based on the choice of the user he land on the requested screen.

Step 5:

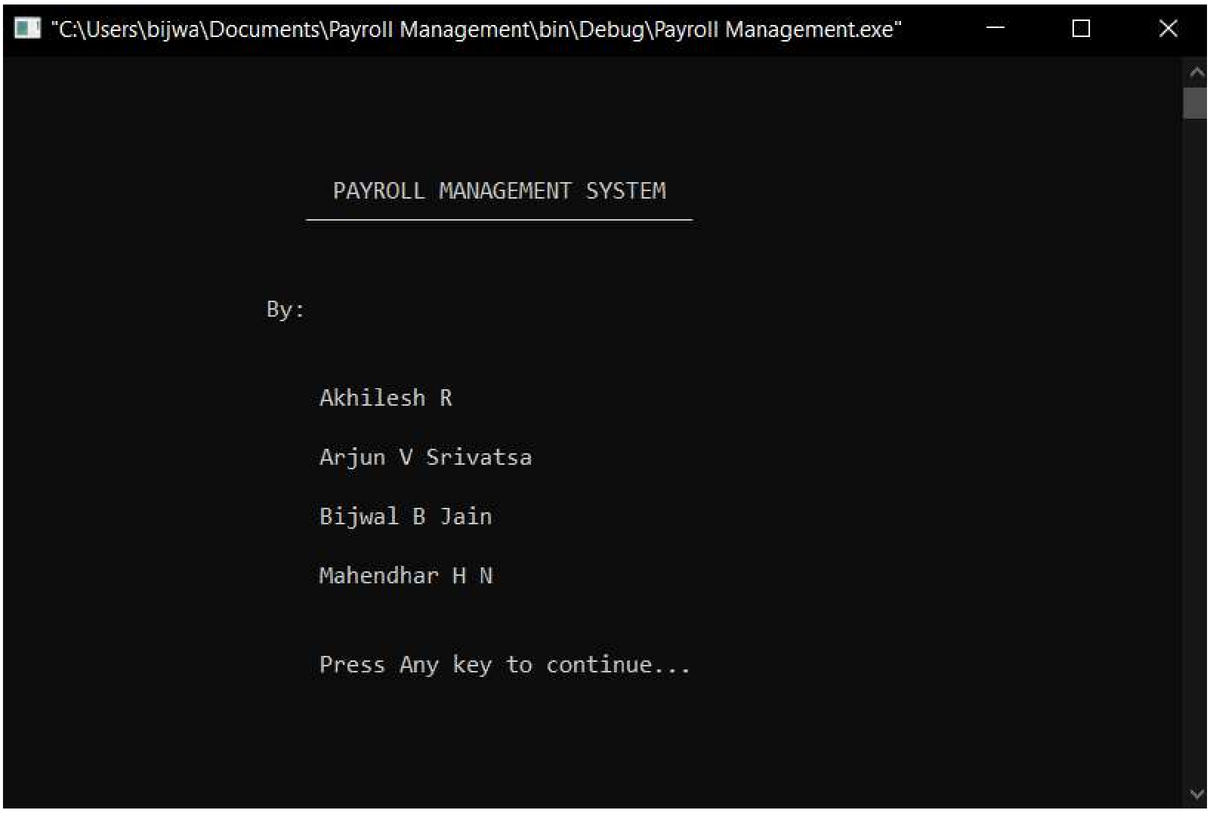
After the job done, all the modified or created data is stored in the files and the software is successfully exited.

Algorithm:

- All required libraries are loaded to the program.
- Incising number of. Employee’s into zero.
- Creating class employee which has many attributes like name, code designation, experience, age, salary etc.
- Initialising function to read the old records by a text, while called as record. Yes, loads all the previous employee record in the record file into the class employee.
- Creating many functions such as Goto, XY etcetera for making graphical interface.
- Function for intro Window which has the names and title.
- Creating a login page through a function for taking admin and password to make it secure.
- Creating a function for taking input from the user for many activities, such as inserting a new record, editing a record directing record, searching a record list in the employee stable and printing employee pay slip.
- For each of the activities mentioned above, a function was created.

- Creating a new record insert function was used which had many attributes like Name, age, working hours, designation etcetera and then it was saved into the class employee.
- For editing a record edit function was used which edits all the attributes mentioned in the creating function. It also alters the pay slip of a person. According to edited value.
- For deleting a record delete function was used which deletes Object. from class employee.
- For searching a record. The user input of employee code was will be taken and searched in the class employee using for loop and ifelse statement.
- List function was created. For listing all the employees.
- There's a function to print a pay slip which prints all the necessary informations about the salary of an employee.
- After all this exit function was created which exit from the GG interface and also storesafter all this exit function was created which exit from the GG interface and also stores back after all, this exit function was created which exit from the Zee Jio interface and also stores back all the information from the class employee Into text mentioned earlier, named as record.

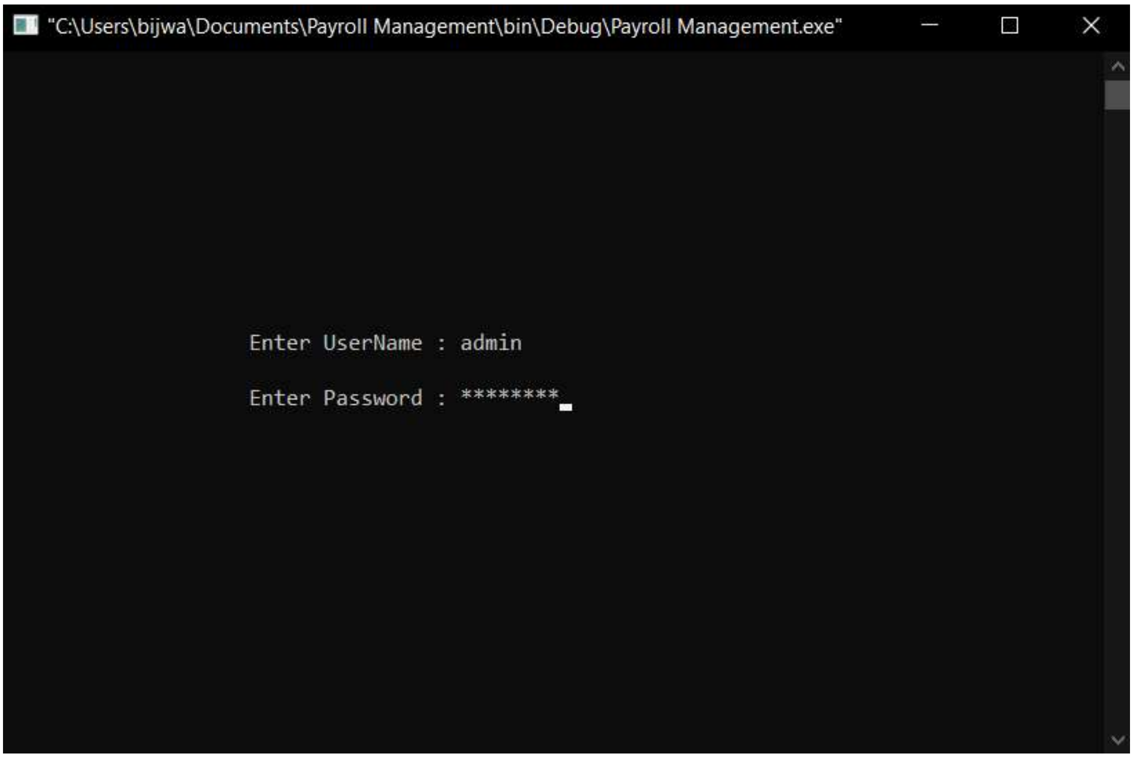
5.Results :



"C:\Users\bijwa\Documents\Payroll Management\bin\Debug\Payroll Management.exe"

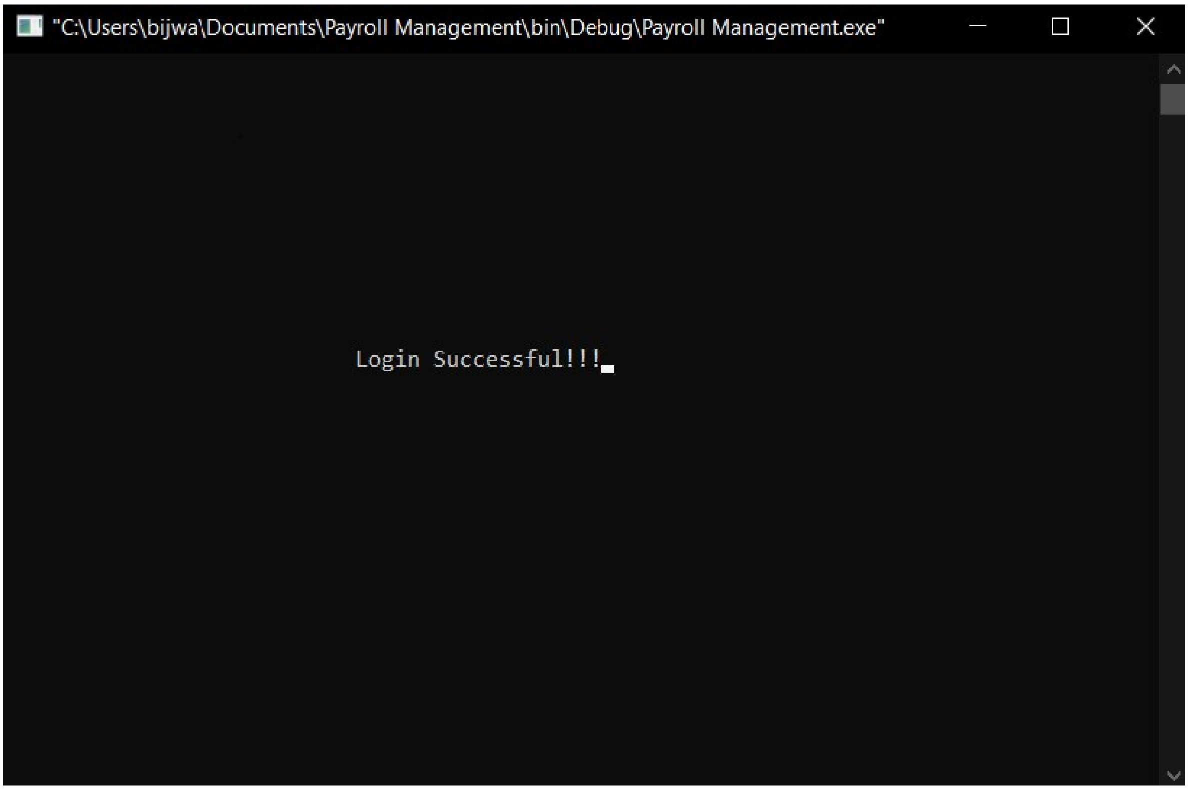
```
                PAYROLL MANAGEMENT SYSTEM
              _____________________________


           By:


               Akhilesh R

               Arjun V Srivatsa

               Bijwal B Jain

               Mahendhar H N


               Press Any key to continue...
```

Figure 5.1: Welcome Screen


"C:\Users\bijwa\Documents\Payroll Management\bin\Debug\Payroll Management.exe"

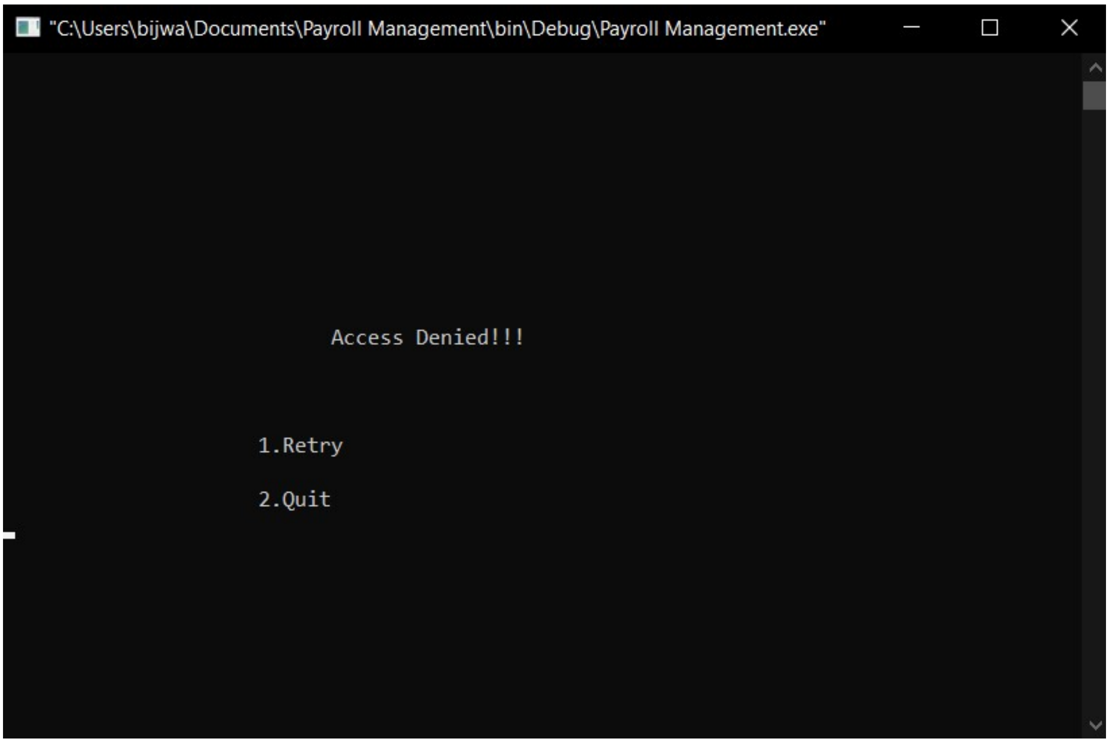
```
Enter UserName : admin

Enter Password : ********
```

Figure 5.2: Login page


"C:\Users\bijwa\Documents\Payroll Management\bin\Debug\Payroll Management.exe"

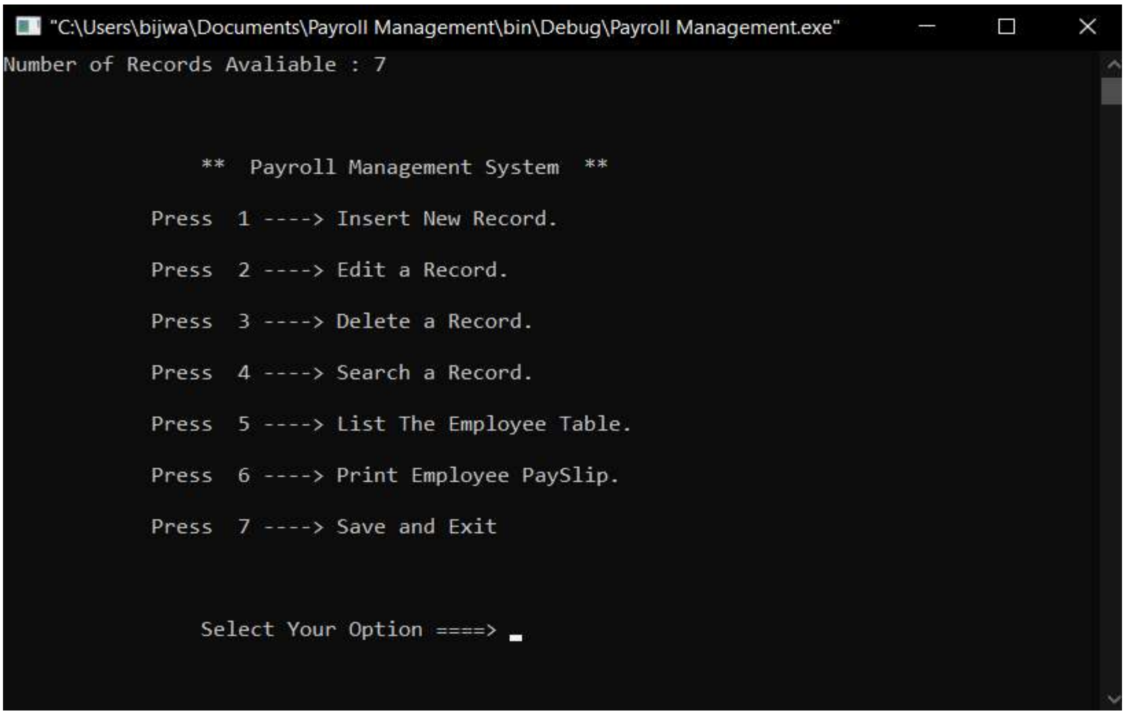
```
Login Successful!!!
```

Figure 5.3: Login Successfull



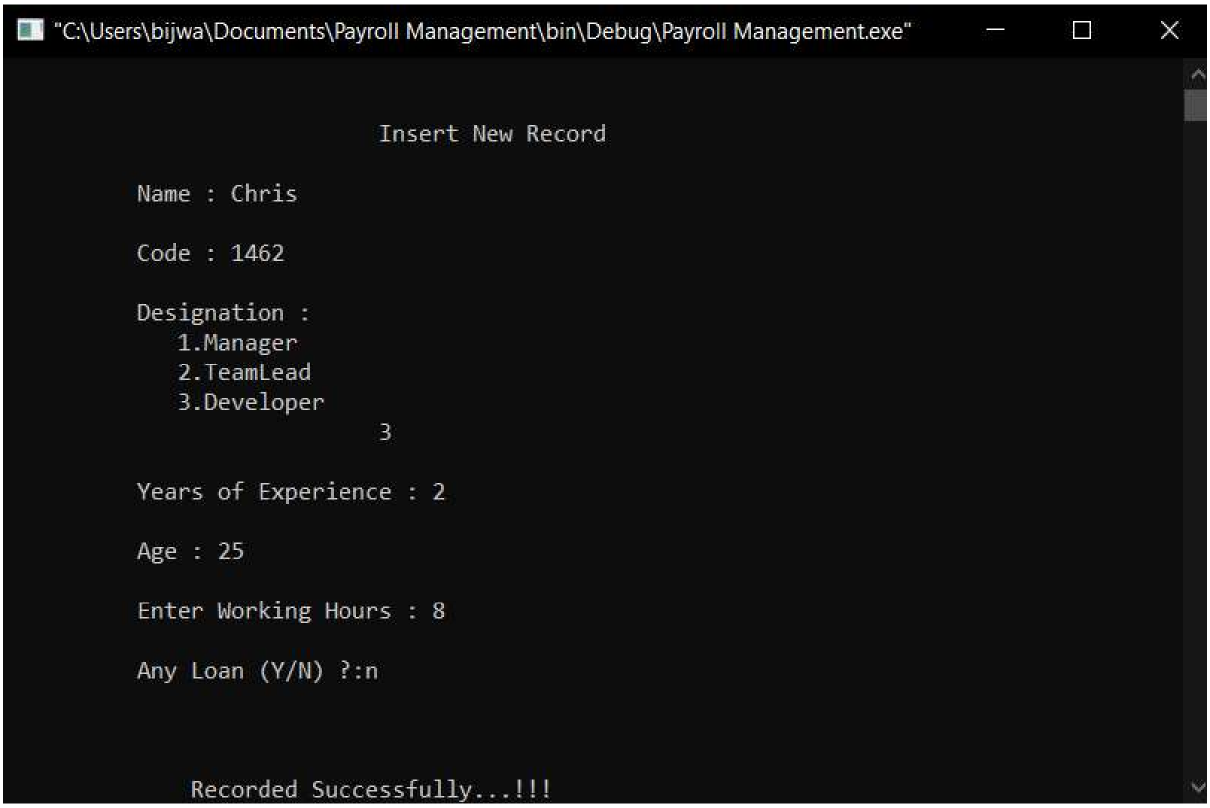
```
"C:\Users\bijwa\Documents\Payroll Management\bin\Debug\Payroll Management.exe"

                        Access Denied!!!


                  1.Retry

                  2.Quit
```

Figure 5.4: Login failed



```
"C:\Users\bijwa\Documents\Payroll Management\bin\Debug\Payroll Management.exe"
Number of Records Avaliable : 7


                **  Payroll Management System  **

            Press  1 ----> Insert New Record.

            Press  2 ----> Edit a Record.

            Press  3 ----> Delete a Record.

            Press  4 ----> Search a Record.

            Press  5 ----> List The Employee Table.

            Press  6 ----> Print Employee PaySlip.

            Press  7 ----> Save and Exit


                Select Your Option ====>
```

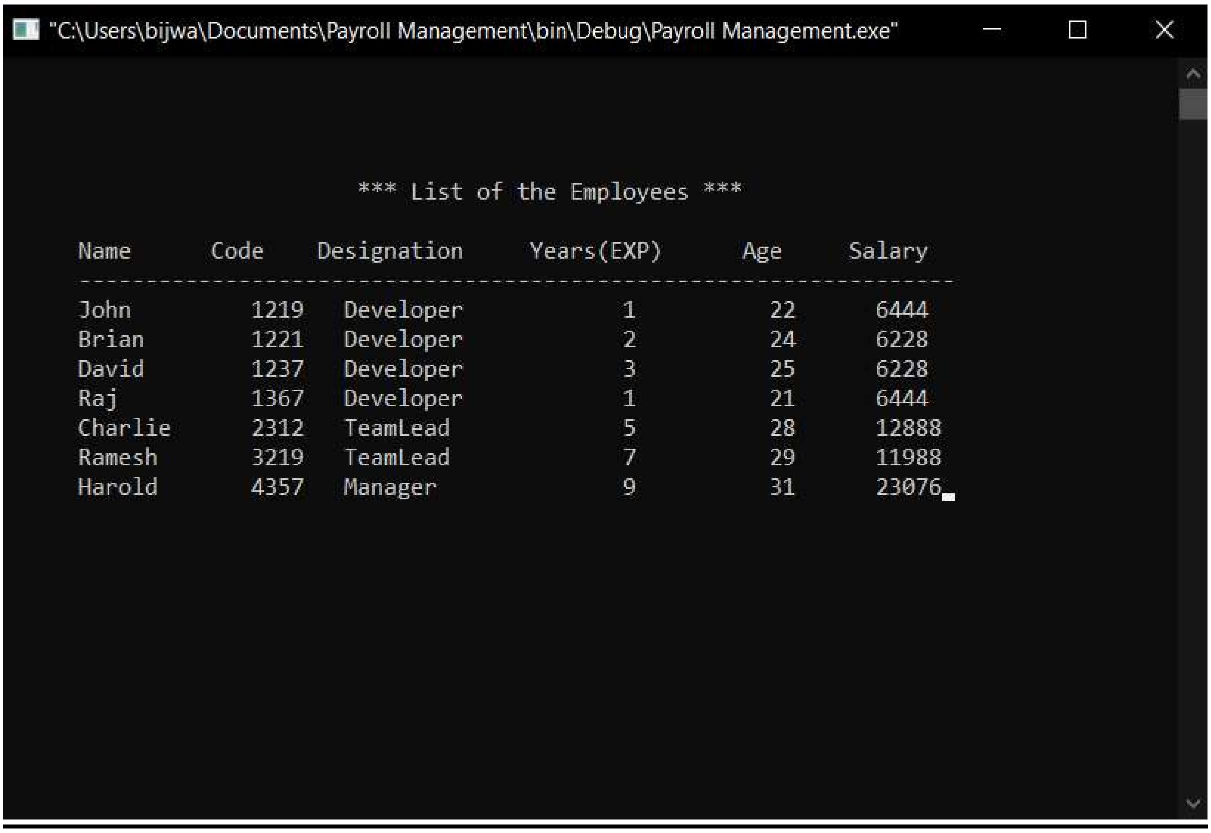
Figure 5.5: Menu page



"C:\Users\bijwa\Documents\Payroll Management\bin\Debug\Payroll Management.exe"

```
                        Insert New Record

        Name : Chris

        Code : 1462

        Designation :
           1.Manager
           2.TeamLead
           3.Developer
                            3

        Years of Experience : 2

        Age : 25

        Enter Working Hours : 8

        Any Loan (Y/N) ?:n


            Recorded Successfully...!!!
```

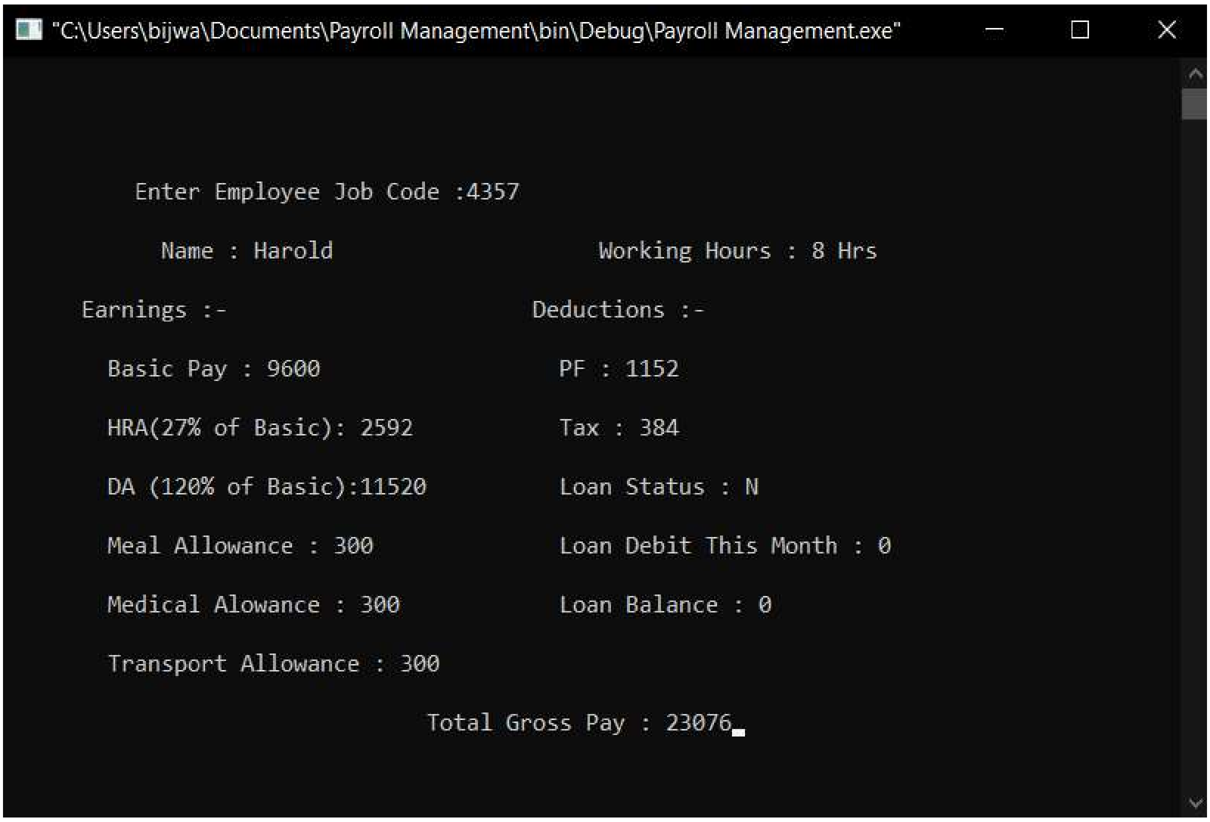
Figure 5.6: Insert new record



"C:\Users\bijwa\Documents\Payroll Management\bin\Debug\Payroll Management.exe"

```
                      *** List of the Employees ***

    Name      Code    Designation     Years(EXP)      Age     Salary
    ------------------------------------------------------------------
    John         1219   Developer             1          22       6444
    Brian        1221   Developer             2          24       6228
    David        1237   Developer             3          25       6228
    Raj          1367   Developer             1          21       6444
    Charlie      2312   TeamLead              5          28       12888
    Ramesh       3219   TeamLead              7          29       11988
    Harold       4357   Manager               9          31       23076
```

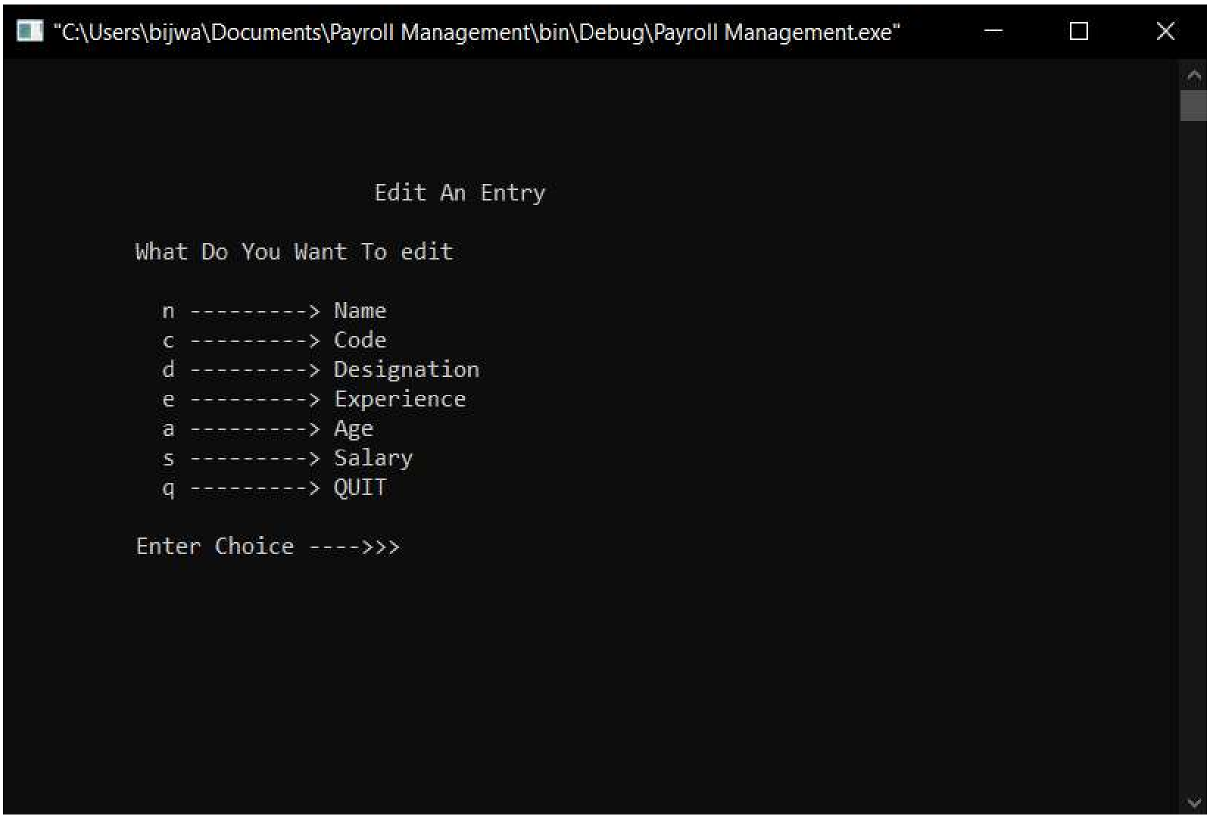
Figure 5.7: Employee list



"C:\Users\bijwa\Documents\Payroll Management\bin\Debug\Payroll Management.exe"

```
     Enter Employee Job Code :4357

       Name : Harold                      Working Hours : 8 Hrs

   Earnings :-                       Deductions :-

     Basic Pay : 9600                   PF : 1152

     HRA(27% of Basic): 2592            Tax : 384

     DA (120% of Basic):11520           Loan Status : N

     Meal Allowance : 300               Loan Debit This Month : 0

     Medical Alowance : 300             Loan Balance : 0

     Transport Allowance : 300

                               Total Gross Pay : 23076
```

Figure 5.8: Payslip



"C:\Users\bijwa\Documents\Payroll Management\bin\Debug\Payroll Management.exe"

```
                       Edit An Entry

     What Do You Want To edit

       n ---------> Name
       c ---------> Code
       d ---------> Designation
       e ---------> Experience
       a ---------> Age
       s ---------> Salary
       q ---------> QUIT

     Enter Choice ---->>>
```

Figure 5.9: Edit menu

6.Importance:

- **Easy to Process**:

Payroll Management System makes the tasks easier. As it requires very less input because it has software in which one has to feed all the information of employee to get the data of calculating wages, finding taxes, paying taxes to the government, financial records which also includes the bonuses, deductions and net pay. These records need to be maintained and recorded by the employer for future references or verification.

- **Performance Management :**

Performance management records the performance information of each and every employee. Hence, it helps the employers to get a clear view related to the performance and skills of their employees which further helps the management in scheduling training according to the needs and requirement of each employee. This type of employee performance review can be highly beneficial when it comes to planning and overall productivity improvement.

- **Optimized Process :**

Payroll System helps companies to reduce the number of employees who are manually working in the maintenance of the attendance, salary, working hours, income taxes etc.. Frequently repeating the payroll tasks sometimes lead to boredom and results in the error. So, it is always advisable to optimize the process so that accurate information is available.

- **Accuracy :**

Payroll Management System help business enterprises to reduce or rather eliminate the errors in the process of payroll. Payroll process is used to get more accurate data. The inconsistency can be easily tackled. It will provide only appropriate and reliable payroll information which minimizes the inefficiency.

7.Advantages:

- Easy and precise calculations.
- Work out payroll calculations and deductions quiker.
- It reduces the burden of compliance.
- Generates accurate payslips.
- Stores data such as payslips and annual reports in a secure, easily accessible system.

8.Applications:

Payroll system is an ideal system and smoothens the HR and payroll process for any organization. It also reduces the possible errors during the performance of payroll system and enhances the system’s performance. Be cautious while choosing any HR Payroll Management system for any organization. Choose the best HR Payroll Management software which is highly secured as it consists of information about your organization and its employees.

9.Conclusion:

The delivered system "PAYROLL MANAGEMENT SYSTEM" software developed for a company has been designed to achieve maximum efficiency and reduce the time taken to handle the payroll activity. It is designed to replace an existing manual record system thereby reducing time taken for calculations and for storing data. The system used C++ as front end.

The system is strong enough to withstand regressive daily operations under conditions where the data is cleaned over a certain time of span. The implementation of the system in the organization will considerably reduce data entry, time and also provide readily calculated reports.

It provides Automation of the entire system improves the efficiency, it provides a friendly Interface which proves to be better when compared to the existing system updating of information becomes so easier.

The System has adequate scope for modification in future if it is necessary.

10.Future Scope:

- Continous tracking of their activities within the company of every employee can be provided with their individual login credintials.
- Auto update of the details like bank loan update can be provided.
- Administration control over software is to be improved.

  Ex: Administration can able to add the employee credentials such that the employee can know their status and companies administration legally.
- Databases can be used for easier and efficient access.
- The proposed system is capable of handling only 100 employee records and can be improved.

11.References:

- https://www.geeksforgeeks.org/object-oriented-programming-in-cpp/
- https://www.entrepreneur.com/encyclopedia/payroll-management
- https://www.farorecruitment.com.vn/definitions-payroll-management-t_1/315
- https://www.greythr.com/complete-guide-payroll/

APPENDIX – SOURCE CODE

```
#include<stdio.h>
#include<iostream>
#include<stdlib.h>
#include<string.h>
#include<conio.h>
#include<windows.h>
#include<time.h>
#include<iomanip>
#include<ctype.h>
#define max 100
using namespace std;

int num=0;
void gotoXY(int,int);
void intro();
void login();
void menu();
void insert();
void edit();
void editmenu();
void editname(int);
void editcode(int);
void editdes(int);
void editexp(int);
void editage(int);
void editsalary(int);
void list();
void deletes();
void search();
void setWindowSize(int,int);
void saverecords();
void getrecords();
bool isFilePresent();
void displayPayslip();
void update_sal();

class employee
{
public:
        char name[20];
        int code;
        char designation[20];
        int exp;
        int age;
        int salary;
        char AnyLoan;
        int HRA;
        int PF;
        int tax;
        int MealAllowance;
        int TransportAllowance;
        int MedicalAllowance;
        int LoanBalance;
        int LoanDebit;
        int grosspay;
        int workingHours;
        int DA;

};
employee emp[max],tempemp[max];
```

APPENDIX – SOURCE CODE

```
void getrecords()
{
    FILE *fp;
    fp = fopen("Records.txt","r");
    int c=0;
    if(fp!=NULL)
    {
        while(feof(fp)==0)
        {

    fscanf(fp,"%s\t%d\t%s\t%d\t%d\t%d\t%c\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n",&emp[c].name,&emp[c].code,&emp[c].designation,&emp[c].exp,&emp[c].age,&emp[c].salary,&emp[c].AnyLoan,&emp[c].HRA,&emp[c].PF,&emp[c].tax,&emp[c].MealAllowance,&emp[c].TransportAllowance,&emp[c].MedicalAllowance,&emp[c].LoanBalance,&emp[c].LoanDebit,&emp[c].grosspay,&emp[c].workingHours,&emp[c].DA);
            c++;
        }
        num=c;
    }
    fclose(fp);
}

void saverecords()
{
    if(num==0)
    {
        system("del Records.txt");
    }
    else
    {
        FILE *fp;
        fp = fopen("Records.txt","w");
        for(int i=0;i<num;i++)
        {

    fprintf(fp,"%s\t%d\t%s\t%d\t%d\t%d\t%c\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n",emp[i].name,emp[i].code,emp[i].designation,emp[i].exp,emp[i].age,emp[i].salary,emp[i].AnyLoan,emp[i].HRA,emp[i].PF,emp[i].tax,emp[i].MealAllowance,emp[i].TransportAllowance,emp[i].MedicalAllowance,emp[i].LoanBalance,emp[i].LoanDebit,emp[i].grosspay,emp[i].workingHours,emp[i].DA);
        }
        fclose(fp);
    }
}

void Cdelay(int msec)
{
  long goal = msec + (clock());
  while (goal > (clock()));
}

bool isFilePresent()
{
    FILE *fp;
    fp = fopen("Records.txt","r");
    if(fp==NULL)
        return false;
    else
        return true;
}
```

APPENDIX – SOURCE CODE

```
void login()
{

           char UserName[30],Password[30],ch;int i=0;
           gotoXY(20,10);
   printf("Enter UserName : ");

   cin>>UserName;
   gotoXY(20,12);
   cout<<"Enter Password : ";
   while(1)
   {
           ch = getch();
           if(ch==13)
                      break;
           if(ch==32||ch==9)
                      continue;
           else
           {
                                 cout<<"*";
                                 Password[i]=ch;
                                 i++;
           }
           }
           Password[i] = '\0';
   if(strcmp(UserName,"admin")==0 && strcmp(Password,"password")==0)
   {
           system("cls");
           gotoXY(27,10);
           cout<<"Login Successful!!!";
           Cdelay(1000);
           }
           else
           {
              int err;
                      system("cls");
           gotoXY(27,10);
                      cout<<"Access Denied!!!\a";
           Cdelay(3000);
           gotoXY(21,14);
           cout<<"1.Retry\n";
           gotoXY(21,16);
           cout<<"2.Quit\n";
           cin>>err;
           if(err==1)
      {
         system("cls");
         login();
      }
      else
      {
         system("cls");
         exit(0);
      }
           }
}
void setWindowSize(int width=670,int height=445)
{
           HWND console = GetConsoleWindow();
           RECT r;
```

APPENDIX – SOURCE CODE

```
            GetWindowRect(console, &r);
            MoveWindow(console, r.left, r.top, width, height, TRUE);
}
void gotoXY(int X, int Y)
{
   COORD coordinates;
   coordinates.X = X;
   coordinates.Y = Y;
   SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}

void update_sal(int i,int h,int desm,char loan,int LoanBal)
{
   int sal,PF,TAX,HRA,MealA,MedicalA,TransportA,DA,LoanDeb;
   sal = h*desm*300;
            TAX =  0.04 * sal;
            DA = 1.20 * sal;
            PF = 0.12 * sal;
            HRA = sal * 0.27;
            MealA = 300;
            MedicalA = 300;
            TransportA = 300;
            LoanDeb = sal * 0.09;
            if(LoanDeb > LoanBal)
            {
                        LoanBal = 0;
                        LoanDeb = LoanBal;
            }
            emp[i].salary = sal;
            emp[i].DA = DA;
            emp[i].tax=TAX;
            emp[i].PF = PF;
            emp[i].HRA = HRA;
            emp[i].MealAllowance = MealA;
            emp[i].MedicalAllowance = MedicalA;
            emp[i].TransportAllowance = TransportA;
            emp[i].LoanBalance = LoanBal-LoanDeb;
            emp[i].AnyLoan = loan;
            emp[i].LoanDebit = LoanDeb;
            emp[i].grosspay = (sal+MealA+MedicalA+TransportA+HRA+DA)-(PF+TAX+LoanDeb) ;
}

void insert()
{

            int i=num;
            int LoanBal,h,des,desm;
            char loan;
            num+=1;
            gotoXY(28,4);
            cout<<"Insert New Record";
            gotoXY(10,6);
            cout<<"Name : ";
            cin>>emp[i].name;
            gotoXY(10,8);
            cout<<"Code : ";
            cin>>emp[i].code;
            desig:
   gotoXY(10,10);
            cout<<"Designation : \n";
```

APPENDIX – SOURCE CODE

```
        cout<<"          1.Manager\n";
        cout<<"          2.TeamLead\n";
        cout<<"          3.Developer\n";
        cout<<"                      ";
        cin>>des;
        switch(des)
  {
    case 1: desm=4;
          strcpy(emp[i].designation,"Manager");
          break;
    case 2: desm=2;
          strcpy(emp[i].designation,"TeamLead");
          break;
    case 3: desm=1;
          strcpy(emp[i].designation,"Developer");
          break;
  }
        gotoXY(10,16);
        cout<<"Years of Experience : ";
        cin>>emp[i].exp;
        gotoXY(10,18);
        cout<<"Age : ";
        cin>>emp[i].age;
        gotoXY(10,20);
        cout<<"Enter Working Hours : ";
        cin>>h;
        emp[i].workingHours = h;
        do
        {
                gotoXY(10,22);
                cout<<"Any Loan (Y/N) ?: \b \b";
                loan=getche();
                if(loan=='Y'|| loan == 'y' || loan =='n' || loan == 'N')
                        break;
        }while(1);
        if(loan=='y'|| loan=='Y')
        {
        gotoXY(10,24);
        cout<<"Enter Loan Balance : ";
        cin>>LoanBal;
        }
        else
        {
                LoanBal=0;
        }
        gotoXY(14,26);
        cout<<"Recorded Successfully...!!!";

        update_sal(i,h,desm,loan,LoanBal);

        getch();
}

void intro()
{
        gotoXY(25,4); cout<<"PAYROLL MANAGEMENT SYSTEM";
  gotoXY(23,5); for(int i=0;i<29;i++) printf("%c",196);
  gotoXY(20,8); cout<<"By:";
        gotoXY(24,11); cout<<"Akhilesh R";
        gotoXY(24,13); cout<<"Arjun V Srivatsa";
```

APPENDIX – SOURCE CODE

```
        gotoXY(24,15); cout<<"Bijwal B Jain";
        gotoXY(24,17); cout<<"Mahendhar H N";
   gotoXY(24,20);cout<<"Press Any key to continue...";
   getch();

}

void  list()
{
        system("cls");
        gotoXY(20,4);
        cout<<"       *** List of the Employees ***";
        gotoXY(6,6);
        cout<<"Name\tCode\tDesignation\tYears(EXP)\tAge\tSalary "<<endl;
        gotoXY(6,7);
        cout<<"---------------------------------------------------------------"<<endl;
        for(int i=0,j=8;i<=num-1;i++,j++)
        {
                gotoXY(6,j);
                cout<<emp[i].name;
                gotoXY(19,j);
                cout<<emp[i].code;
                gotoXY(26,j);
                cout<<emp[i].designation;
                gotoXY(47,j);
                cout<<emp[i].exp;
                gotoXY(58,j);
                cout<<emp[i].age;
                gotoXY(66,j);
                cout<<emp[i].grosspay;
        }
        getch();
}


void menu()
{
        system("cls");
        gotoXY(0,0);
        cout<<"Number of Records Avaliable : "<<num;
        gotoXY(16,4);
        cout<<"**  Payroll Management System  ** ";
        gotoXY(12,6);
        cout<<"Press  1 ----> Insert New Record.";
        gotoXY(12,8);
        cout<<"Press  2 ----> Edit a Record.";
        gotoXY(12,10);
        cout<<"Press  3 ----> Delete a Record.";
        gotoXY(12,12);
        cout<<"Press  4 ----> Search a Record.";
        gotoXY(12,14);
        cout<<"Press  5 ----> List The Employee Table.";
        gotoXY(12,16);
        cout<<"Press  6 ----> Print Employee PaySlip.";
        gotoXY(12,18);
        cout<<"Press  7 ----> Save and Exit";
        gotoXY(16,22);
        cout<<"Select Your Option ====> ";
}
```

APPENDIX – SOURCE CODE

```
void deletes()
{
        for(int i=0;i<num;i++)
        {
                tempemp[i]=emp[i];
        }
        //system("cls");
        int code;
        int check=-1;
        gotoXY(28,4);
        cout<<"Delete a Record";
        gotoXY(10,6);
        cout<<"Enter the JobCode To Delete That Record  :";
        cin>>code;
        int i,j;
        for(i=0;i<=num-1;i++)
        {
                if(emp[i].code==code)
                {
                        check=i;
                }
        }
        if(check!=-1)
        {
                for(i=0,j=0;i<=num-1;i++,j++)
                {
                        if(i==check)
                        {
                                i++;
                        }
                        emp[j]=tempemp[i];
                }
                num--;
        }
}

void search()
{
        system("cls");
        int jobcode;
        bool found = false;
        gotoXY(10,4);
        cout<<"You can Search only through the Jobcode of an Employee";
        gotoXY(10,6);
        cout<<"Enter Code Of the Employee : ";
        cin>>jobcode;
        for(int i=0;i<=num-1;i++)
        {
                if(emp[i].code==jobcode)
                {
                        gotoXY(6,8);
                        cout<<"Name\tCode\tDesignation\tYears(EXP)\tAge\tSalary "<<endl;
                        gotoXY(6,9);
                        cout<<"---------------------------------------------------------------"<<endl;
                        gotoXY(6,11);
                        cout<<emp[i].name;
                        gotoXY(19,11);
                        cout<<emp[i].code;
                        gotoXY(26,11);
                        cout<<emp[i].designation;
```

APPENDIX – SOURCE CODE

```
                gotoXY(47,11);
                cout<<emp[i].exp;
                gotoXY(58,11);
                cout<<emp[i].age;
                gotoXY(66,11);
                cout<<emp[i].grosspay;
                found = true;
            }
        }
        if(!found)
        {
            gotoXY(26,11);
            cout<<"No records Found...!!!\a";
        }
        getch();
}

void editmenu()
{
  system("cls");
        gotoXY(28,4);
        printf("Edit An Entry");
        gotoXY(10,6);
        cout<<"What Do You Want To edit";
        gotoXY(12,8);
        cout<<"n ---------> Name ";
        gotoXY(12,9);
        cout<<"c ---------> Code ";
        gotoXY(12,10);
        cout<<"d ---------> Designation";
        gotoXY(12,11);
        cout<<"e ---------> Experience ";
        gotoXY(12,12);
        cout<<"a ---------> Age";
        gotoXY(12,13);
        cout<<"s ---------> Salary";
        gotoXY(12,14);
        cout<<"q ---------> QUIT";
        gotoXY(10,16);
        cout<<"Enter Choice ---->>>  ";
}

void editname(int i)
{
        gotoXY(10,18);
        cout<<"Enter New Name----->  ";
        cin>>emp[i].name;
}

void editcode(int i)
{
        gotoXY(10,18);
        cout<<"Enter New Job Code----->  ";
        cin>>emp[i].code;
}
void editdes(int i)
{
  int des,desm;
        gotoXY(10,18);
        cout<<"Choose new designation----->\n";
```

APPENDIX – SOURCE CODE

```
        cout<<"                    1.Manager\n";
        cout<<"                    2.TeamLead\n";
        cout<<"                    3.Developer\n";
        cout<<"                              ";
        cin>>des;
        switch(des)
  {
    case 1: desm=4;
         strcpy(emp[i].designation,"Manager");
         break;
    case 2: desm=2;
         strcpy(emp[i].designation,"TeamLead");
         break;
    case 3: desm=1;
         strcpy(emp[i].designation,"Developer");
         break;
  }
        update_sal(i,emp[i].workingHours,desm,emp[i].AnyLoan,emp[i].LoanBalance);
}

void editexp(int i)
{
        gotoXY(10,18);
        cout<<"Enter new Years of Experience";
        cin>>emp[i].exp;
}
void editage(int i)
{
        gotoXY(10,18);
        cout<<"Enter new Age ";
        cin>>emp[i].age;
}

void editsalary(int i)
{
        int sal,PF,TAX,HRA,MealA,MedicalA,TransportA,LoanBal=emp[i].LoanBalance,LoanDeb,DA;
        char loan;
        gotoXY(10,18);
        cout<<"Enter new Salary ";
        cin>>sal;
        DA = 1.20 * sal;
        TAX =  0.04 * sal;
        PF = 0.12 * sal;
        HRA = 4000;
        MealA = 300;
        MedicalA = 300;
        TransportA = 300;
        LoanDeb = sal * 0.09;
        if(LoanDeb > LoanBal)
        {
                LoanBal = 0;
                LoanDeb = LoanBal;
        }
        emp[i].salary = sal;
        emp[i].tax=TAX;
        emp[i].PF = PF;
        emp[i].HRA = HRA;
        emp[i].MealAllowance = MealA;
        emp[i].MedicalAllowance = MedicalA;
        emp[i].TransportAllowance = TransportA;
```

APPENDIX – SOURCE CODE

```
    emp[i].LoanBalance = LoanBal;
    emp[i].AnyLoan = loan;
    emp[i].LoanDebit = LoanDeb;
    emp[i].grosspay = (sal+MealA+MedicalA+TransportA+HRA+DA)-(PF+TAX+LoanDeb) ;
}

void edit()
{
    system("cls");
    int jobcode;
    gotoXY(28,4);
    printf("Edit a Record");
    int i;
    char option;
    gotoXY(10,6);
    cout<<"Enter the jobcode To Edit : ";
    cin>>jobcode;
    editmenu();
    for(i=0;i<=num-1;i++)
    {
        if(emp[i].code==jobcode)
        {
            while((option=cin.get())!='q')
            {
                switch(option)
                {
                    case 'n':
                        editname(i);
                        break;
                    case 'c':
                        editcode(i);
                        break;
                    case 'd':
                        editdes(i);
                        break;
                    case 'e':
                        editexp(i);
                        break;
                    case 'a':
                        editage(i);
                        break;
                    case 's':
                        editsalary(i);
                        break;
                }
                editmenu();
            }
        }
    }
}


void displayPayslip()
{
    system("cls");
    gotoXY(10,4);
    int code,i;
    cout<<"Enter Employee Job Code :";
    cin>>code;
    for(i=0;i<=num-1;i++)
```

```
        {
                if(emp[i].code==code)
                {
                        gotoXY(12,6);
                        cout<<"Name : "<<emp[i].name;
                        gotoXY(45,6);
                        cout<<"Working Hours : "<<emp[i].workingHours<<" Hrs";
                        gotoXY(6,8);
                        cout<<"Earnings :-";
                        gotoXY(8,10);
                        cout<<"Basic Pay : "<<emp[i].salary<<endl;
                        gotoXY(8,12);
                        cout<<"HRA(27% of Basic): "<<emp[i].HRA<<endl;
                        gotoXY(8,14);
                        cout<<"DA (120% of Basic):"<<emp[i].DA;
                        gotoXY(8,16);
                        cout<<"Meal Allowance : "<<emp[i].MealAllowance<<endl;
                        gotoXY(8,18);
                        cout<<"Medical Alowance : "<<emp[i].MedicalAllowance<<endl;
                        gotoXY(8,20);
                        cout<<"Transport Allowance : "<<emp[i].TransportAllowance<<endl;
                        gotoXY(40,8);
                        cout<<"Deductions :- "<<endl<<endl;
                        gotoXY(42,10);
                        cout<<"PF : "<<emp[i].PF<<endl;
                        gotoXY(42,12);
                        cout<<"Tax : "<<emp[i].tax<<endl;
                        gotoXY(42,14);
                        int l = emp[i].AnyLoan;
                        char l2 = toupper(l);
                        cout<<"Loan Status : "<<l2<<endl;
                        gotoXY(42,16);
                        cout<<"Loan Debit This Month : "<<emp[i].LoanDebit<<endl;
                        gotoXY(42,18);
                        cout<<"Loan Balance : "<<emp[i].LoanBalance<<endl;
                        gotoXY(32,22);
                        cout<<"Total Gross Pay : "<<emp[i].grosspay;
                }
        }
        getch();
}




int main()
{
        setWindowSize();
        intro();
        system("CLS");
  login();
  system("CLS");
  getrecords();
  menu();
  char option;
  if(emp[0].code==0 && isFilePresent())
        num--;
        while(1)
        {
```

APPENDIX – SOURCE CODE

```
            option=getch();
            system("CLS");
            switch(option)
            {
                case '1':
                    insert();
                    break;
                case '2':
                    edit();
                    break;
                case '3':
                    deletes();
                    break;
                case '4':
                    search();
                    break;
                case '5':
                    list();
                    break;
                case '6':
                    displayPayslip();
                    break;
                case '7':
                    saverecords();
                    exit(0);


            }
            system("CLS");
            menu();
        }
        return 0;
}
```