**Kathmandu University**

**Department of Computer Science & Engineering**

**B. tech Artificial Intelligence**

**Year- I/II**

Panchkhal, Kathmandu



**"Payroll Management System"**

**Submitted by: Submitted to:**

Jayed Alam Mansur --------------------

Roll No.: 18 2nd Semester

B. tech AI (I/II)

Email: jayedalamsigningoff@gmail.com

Date: \_\_\_\_\_\_\_\_\_\_2024

# DECLARATION

I hereby declare that the project work entitled **"Payroll Management System"** submitted to Department of Computer Science & Engineering, Kathmandu University, Dhulikhel. **B. tech AI**, Panchkhal, is for the degree is submitted in partial fulfillment of requirements for the degree of Bachelor of Technology in Artificial Intelligence (B. tech AI). This report contains all the original research and analysis with some information taken from other resources which are properly cited. This project work has not been submitted to any other university or institution for the award of any degree or diploma.

**……………………………………………**

**Jayed Alam Mansur**

**Date: \_\_\_\_\_\_\_\_\_ 2024**

# ACKNOWLEDGEMENT

I would like express my deep gratitude to **\_\_\_\_** for providing me this report on the topic **"Payroll Management System"**. It is my immense pleasure to acknowledge and express my heartfelt appreciation to \_\_\_\_\_ for constructive support and guidance for the preparation of this project work.

I am also thankful to the lab teacher **\_\_\_\_\_\_\_**for his help during the completion of the project. I would like to present my teachers gratitude for their guidance, good advices and honest reviews in this entire project session. Project has been completed under the intensive and genuine guidance of \_\_\_\_\_\_.

**Jayed Alam Mansur**

**B. tech AI, 2nd Semester**

**Department of Computer Science & Engineering**

# ABSTRACT

C++ helps you to understand the internal architecture of a computer, how computer stores and retrieves information. After learning C++, it will be much easier to learn other programming languages like Java, Python, etc. [1]

The Payroll **Management System** is implemented by [C++ programming](https://www.geeksforgeeks.org/c/) along with data structure and algorithm The Payroll Management System deals with the financial aspects of employee's salary, allowances, deductions, gross pay, net pay etc. and generation of pay-slips for a specific period. The outstanding benefit of Payroll Management System is its easy implement other advantages of Payroll Management System and its extensive features and reports.The total number of employees will be displayed and an end result of their result is sorted using bubble sort.

# TABLE OF CONTENTS

[DECLARATION i](#_Toc164929062)

[ACKNOWLEDGEMENT ii](#_Toc164929063)

[ABSTRACT iii](#_Toc164929064)

[TABLE OF CONTENTS iv](#_Toc164929065)

[LIST OF FIGURES vi](#_Toc164929066)

[1. INTRODUCTION 1](#_Toc164929067)

[**1.1** **INTRODUCTION TO PAYROLL SYSTEM** 1](#_Toc164929068)

[**1.2** **BACKGROUND OF THE REPORT** 1](#_Toc164929069)

[**1.3** **PROBLEM ANALYSIS** 1](#_Toc164929070)

[**1.4** **OBJECTIVES** 2](#_Toc164929071)

[**1.5** **SCOPE** 2](#_Toc164929072)

[2. STRUCTURE OF THE PROGRAM 3](#_Toc164929073)

[**2.1** **ARCHITECTURE** 3](#_Toc164929074)

[**2.2** **MODULES** 4](#_Toc164929075)

[3. TECHNOLOGIES USED 5](#_Toc164929076)

[**3.1 C++** 5](#_Toc164929077)

[**3.2 C++ Classes and Objects** 5](#_Toc164929078)

[**3.3 C++ FUNCTIONS** 6](#_Toc164929079)

[**3.4 FILE HANDLING IN C++** 7](#_Toc164929080)

[***Opening a File*** 7](#_Toc164929081)

[***Closing a File*** 7](#_Toc164929082)

[***Writing to a File*** 7](#_Toc164929083)

[***Reading from a File*** 7](#_Toc164929084)

[**3.5** **BUBBLE SORT** 7](#_Toc164929085)

[4. IMPLEMENTATION 8](#_Toc164929086)

[**4.1 PROBLEM STATEMENT** 8](#_Toc164929087)

[**4.2 WORKING PROCEDURE** 8](#_Toc164929088)

[**4.3 PROJECT ADT** 9](#_Toc164929089)

[**4.4 SOURCE CODE** 10](#_Toc164929090)

[5. RESULTS 25](#_Toc164929091)

[6. CONCLUSION 29](#_Toc164929092)

[7. REFERENCES 29](#_Toc164929093)

# LIST OF FIGURES

[Figure 1: Flowchart of the project 3](file:///C:\Users\USER\Desktop\Saga_Sem\3rd\dsa_project\Final\National%20College%20of%20Computer%20Studies.docx#_Toc164929094)

[Figure 2: Intro and login page 25](file:///C:\Users\USER\Desktop\Saga_Sem\3rd\dsa_project\Final\National%20College%20of%20Computer%20Studies.docx#_Toc164929095)

[Figure 3: Login Successfully 25](file:///C:\Users\USER\Desktop\Saga_Sem\3rd\dsa_project\Final\National%20College%20of%20Computer%20Studies.docx#_Toc164929096)

[Figure 4: Main Menu 26](file:///C:\Users\USER\Desktop\Saga_Sem\3rd\dsa_project\Final\National%20College%20of%20Computer%20Studies.docx#_Toc164929097)

[Figure 5: Inserting an employee record 26](file:///C:\Users\USER\Desktop\Saga_Sem\3rd\dsa_project\Final\National%20College%20of%20Computer%20Studies.docx#_Toc164929098)

Figure 6: Searching an employee's detail 27

[Figure 7: Unsorted List of employees 27](file:///C:\Users\USER\Desktop\Saga_Sem\3rd\dsa_project\Final\National%20College%20of%20Computer%20Studies.docx#_Toc164929099)

[Figure 8: Sorted List of employees 27](file:///C:\Users\USER\Desktop\Saga_Sem\3rd\dsa_project\Final\National%20College%20of%20Computer%20Studies.docx#_Toc164929099)

Figure 9: Editing an employee's detail 28

Figure 10: Pay Slip of an employee 28

# INTRODUCTION

## **INTRODUCTION TO PAYROLL SYSTEM**

Software called a payroll system is used to arrange all of the administrative work involved in paying employees and reporting their taxes. One of these responsibilities may be tracking hours. Calculating wages, withholding taxes and deductions, printing and delivering checks and paying employment taxes to the government.

Payroll software frequently only needs very minimal wage data to be supplied; after that, it computes the data and handles withholdings automatically. The majority of payroll software will notify employers when certain tax forms need to be filed and will update automatically anytime a tax defect changes.[1]

## **BACKGROUND OF THE REPORT**

The Payroll Management System deals with the financial aspects of employee's salary, allowances, deductions, gross pay, net pay etc. and generation of pay-slips for a specific period. The outstanding benefit of Payroll Management System is its easy implement other advantages of Payroll Management System and its extensive features and reports.[3]

## **PROBLEM ANALYSIS**

While choosing a payroll system can be challenging, there are a few things to consider. First, determine your company's size and the amount of money you are willing to spend on payroll processing.

Smaller companies can handle payroll tasks internally through a manual process, but this can be time-consuming and error-prone. If an employer makes a mistake, they could face financial or legal repercussions. Mid-sized companies with up to employers benefit from payroll system investments.[3]

## **OBJECTIVES**

The key objectives of payroll management system are given below

* Store and manage employee information effectively
* Define the employee's earnings, leaves, etc.
* Generate pay-slip according to salary structure assigned to employee
* Generate all information and reports related to employee

## **SCOPE**

* Recurring payroll services
* Full range of Social Insurance management
* Tasks related to newcomers and departing employees
* Annual services connected to payroll processing and social security administration
* Preparation and submission of tax declarations
* Payroll disbursement services[3]

# STRUCTURE OF THE PROGRAM

## **ARCHITECTURE**

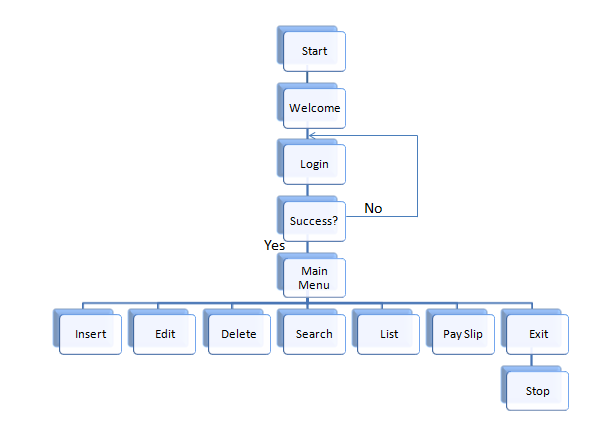
****The following flowchart shows the structure of how the project was made from the starting point of admin information to displaying the employee records.

Figure 1: Flowchart of the project

## **MODULES**

There are three major modules used in this system. The first being admin page that include the login credentials and data entry module. Lastly storing and retrieving data module is the third module used in the project.

**2.2.1 Admin Login Module**

Here in this module call, user is prompted to enter the login credentials. The Login Module is a portal that allows users to type a user name and password to login. This module is no longer available to users after they have logged in. The Login Module appears to users next to the introduction module.

**2.2.2 Data Entry Module**

After you selected data entry from the main menu you land on this screen. In this module, Data of the employee are inserted. The Fields required here are Name, Id of the employee, Designation, Age, Years of experience, No. of working hours, Loan Status if any.

All the required data is processed and the salary, earnings and deductions of the employee are calculated and finally stored in the files for permanent storage.

**2.2.3 Storing and Retrieving Data Records Module**

Records of all the employees are to be maintained and the records are stored in Files and the information is retrieved from the files. All the Records are separated by new lines, and each field of an individual record is separated by ‘tab’.[3]

# TECHNOLOGIES USED

## **3.1 C++**

C++ is a high-level language that has developed in mid of 1970’s. It was created using C. Because of its object-oriented features, programmers can build objects directly within the code. Programming is made simpler and more efficient in terms of both time and space by the creation of objects. Power and flexibility over the language are provided by C++.   
These days, C++ is a popular general-purpose programming language for competitive programming. Its programming features include imperative, object-oriented, and generic. Many platforms, including Windows, Linux, Unix, Mac, and others, support C++.[1]

## **3.2 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. 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.

## **3.3 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.[2]

## **3.4 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.

## ***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.

## **BUBBLE SORT**

Bubble Sort is the simplest sorting algorithm that is used in this project by repeatedly swapping the adjacent job-codes of employees if they are in the wrong order. This is done in ascending order thus employee codes is used for this. This is the only major DSA operation (algorithm) used in this project. This algorithm is used in list() function to sort the employee details in ascending order.

# IMPLEMENTATION

## **4.1 PROBLEM STATEMENT**

It may be difficult to decide which system to choose, but there are some factors to keep in mind when deciding. First, analyze the size of your business and decide how much you are willing to spend on payroll processing.

While it is possible for smaller businesses to handle payroll duties in-house through a manual process, much time can be wasted while attempting to calculate everything correctly. One miscalculation and the business owner could find themselves in legal or financial trouble. Mid-sized companies with up to employers benefit, by investing in a payroll system.

## **4.2 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.[2]

## **4.3 PROJECT ADT**

Here are all the functions used in the project.

void gotoXY(int,int); - Sets the cursor position to the specified location.

void Cdelay(int); - Adds the time delay of specified milliseconds.

void intro(); - Introduction of our project is displayed here.

void login(); - Login Authentication goes here.

void menu(); - Main Menu.

void insert(); - Creates a new record.

void edit(); - Edit a record.

void editmenu(); - Display edit options.

void editname(int); - Employee Name will be edited.

void editcode(int); - Employee Code will be edited.

void editdes(int); - Employee Designation will be edited.

void editexp(int); - Employee Experience will be edited.

void editage(int); - Employee age will be edited.

void editsalary(int); - Employee salary will be edited.

void list(); - Lists all the records available.

void deletes(); - Delete a specific record.

void search(); - Search for the record.

void setWindowSize(int,int); - Set Output window to desired size.

void saverecords(); - All the records created/modified will be saved.

void getrecords(); - All the available records will be retrived.

bool isFilePresent(); - Checks weather the Database File is present or not.

void displayPayslip(); - Displays the Pay Slip of the specified employee.

void bubblesort(); - Sort the employee record via their jobcode[1]

## **4.4 SOURCE CODE**

**CLASS:**

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];

**BUBBLE SORT:**

void bubbleSort(employee arr[], int n) {

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (arr[j].code > arr[j + 1].code)

swap(arr[j], arr[j + 1]);

}

}

}

**GETTING AND SAVING RECORDS:**

void getrecords(){

FILE \*fp;

fp = fopen("Detail.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 Detail.txt");

}

else{

FILE \*fp;

fp = fopen("Detail.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);

}

}

**CDELAY AND GOTO:**

void Cdelay(int msec){

long goal = msec + (clock());

while (goal > (clock()));

}

void gotoXY(int X, int Y){

COORD coordinates;

coordinates.X = X;

coordinates.Y = Y;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coordinates);

}

**LOGIN:**

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 Success!!!";

gotoXY(21,12);

cout<<"Will be redirected in 3 Seconds...";

gotoXY(56,12);

Cdelay(1000);

gotoXY(44,12);

cout<<"\b \b2";

gotoXY(56,12);

Cdelay(1000);

gotoXY(44,12);

cout<<"\b \b1";

gotoXY(56,12);

Cdelay(1000);

}

else

{

system("cls");

gotoXY(27,10);

printf("Access Denied!!!\a");

gotoXY(21,12);

cout<<"Will be redirected in 3 Seconds...";

gotoXY(56,12);

Cdelay(1000);

gotoXY(44,12);

cout<<"\b \b2";

gotoXY(56,12);

Cdelay(1000);

gotoXY(44,12);

cout<<"\b \b1";

gotoXY(56,12);

Cdelay(1000);

system("cls");

login();

}

}

**INSERTING A RECORD:**

void insert(){

system("CLS");

int i=num;

int sal,PF,TAX,HRA,MealA,MedicalA,TransportA,LoanBal,LoanDeb,h,DA;

char loan;

num+=1;

gotoXY(28,4);

printf("Insert New Record");

gotoXY(10,6);

cout<<"Name : ";

cin>>emp[i].name;

gotoXY(10,8);

cout<<"Code : ";

cin>>emp[i].code;

gotoXY(10,10);

cout<<"Designation : ";

cin>>emp[i].designation;

gotoXY(10,12);

cout<<"Years of Experience : ";

cin>>emp[i].exp;

gotoXY(10,14);

cout<<"Age : ";

cin>>emp[i].age;

gotoXY(10,16);

cout<<"Enter Working Hours : ";

cin>>h;

sal = h\*300;

emp[i].workingHours = h;

do{

gotoXY(10,18);

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(10,20);

cout<<"Enter Loan Balance : ";

cin>>LoanBal;

}

else

LoanBal=0;

gotoXY(14,22);

cout<<"Recorded Succesfully...!!!";

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) ;

getch();

}

**LIST OF RECORDS:**

void list(int num)

{

system("PAUSE");

system("cls");

gotoXY(20,4);

printf(" \*\*\*\*\*\*\*\* 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();

}

**MAIN MENU:**

void menu(){

system("cls");

cout<<"Number of Records Avaliable : "<<num;

gotoXY(16,4);

printf("\*\*\*\*\* Payroll Management System \*\*\*\*\* ");

gotoXY(12,6);

cout<<"Press i : Insert New Record.";

gotoXY(12,8);

cout<<"Press e : Edit a Record.";

gotoXY(12,10);

cout<<"Press d : Delete a Record.";

gotoXY(12,12);

cout<<"Press s : Search a Record.";

gotoXY(12,14);

cout<<"Press l : List The Employee Table.";

gotoXY(12,16);

cout<<"Press p : Print Employee PaySlip.";

gotoXY(12,18);

cout<<"Press q : Quit Program.";

gotoXY(16,22);

cout<<"Select Your Option -> ";

}

**DELETING A RECORD:**

void deletes(){

for(int i=0;i<num;i++) {

tempemp[i]=emp[i];

}

system("cls");

int code;

int check=-1;

gotoXY(28,4);

printf("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--;

}

}

**SEARCHING A RECORD:**

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;

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();

**}**

**EDITING A RECORD:**

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){

gotoXY(10,18);

cout<<"enter new designation-----> ";

cin>>emp[i].designation;

}

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;

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;

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();

}

}

}

}

**PAY SLIP OF EMPLOYEE:**

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();

}

**MAIN FUNCTION:**

int main(){

int num=3;

intro();

login();

menu();

getrecords();

char option;

if(emp[0].code==0 && isFilePresent())

num--;

while(1){

cin>>option;

switch(option){

case 'l':

system("CLS");

cout << "Unsorted Employee Records:" << endl;

list(num);

cout << endl;

bubbleSort(emp, num);

system("CLS");

cout << "Sorted Employee Records by Job Code:" << endl;

list(num);

break;

case 'i':

insert();

break;

case 'd':

deletes();

break;

case 'e':

edit();

break;

case 's':

search();

break;

case 'p':

displayPayslip();

break;

case 'q':

saverecords();

exit(0);

}

menu();

}

return 0;

}

# RESULTS

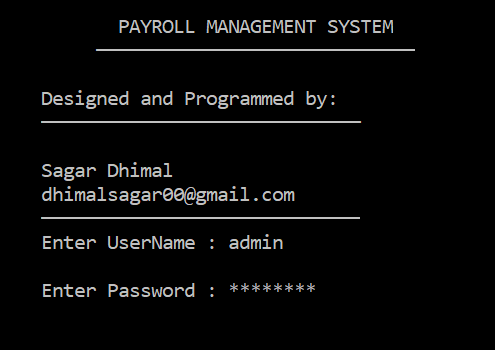


Figure 2: Intro and login page

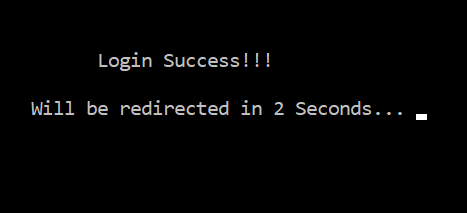


Figure 3: Login Successfully



Figure 4: Main Menu



Figure 5: Inserting an employee record

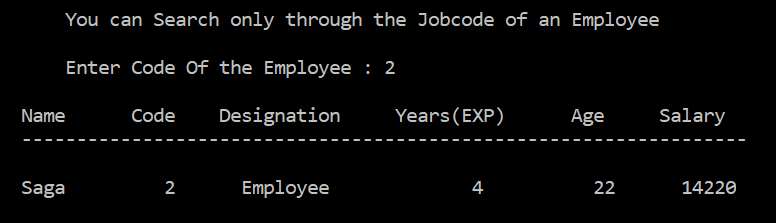
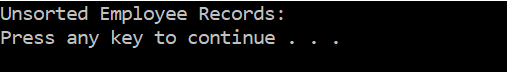
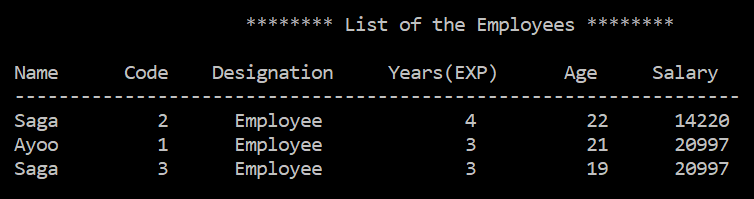
SORTING EMPLOYEE DETAILS USING

Figure 7: Unsorted List of employees

Figure 6: Searching an employee's detail

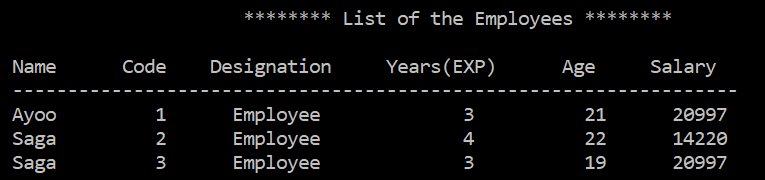
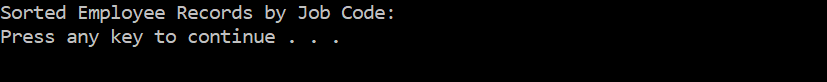
**BUBBLE SORT**

Figure 8: Sorted List of employees

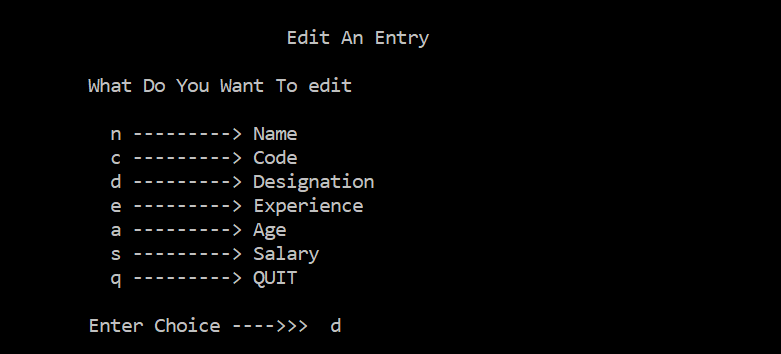
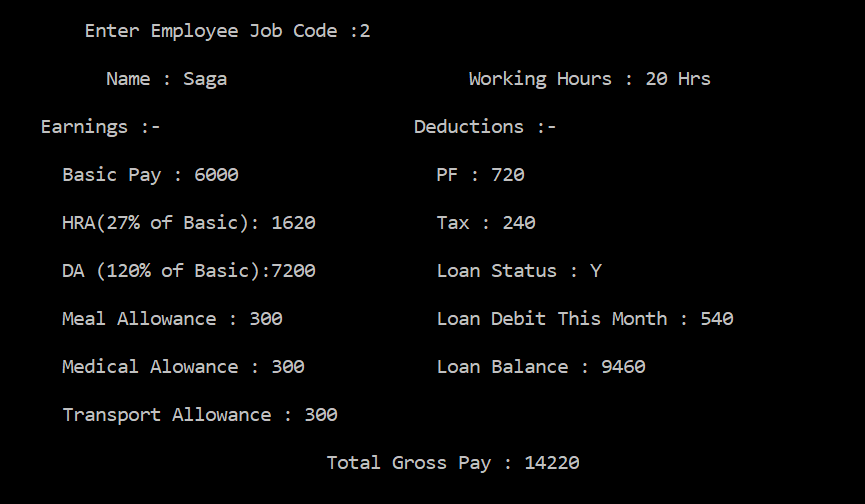


Figure 10: Pay Slip of an employee

Figure 9: Editing an employee's detail

# 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. [3]

# REFERENCES

1. "Geeksforgeeks," [Online]. Available: https://www.geeksforgeeks.org/data-structure-and-algorithm/ [Accessed 25 04 2024].
2. "OpenAI," Open AI, 24 April 2024. [Online]. Available: https://chat.openai.com. [Accessed 24 04 2024]
3. "GitHub" [Online]. Available: https://www.github.com/payroll-system/ [Accessed 24 04 2024].