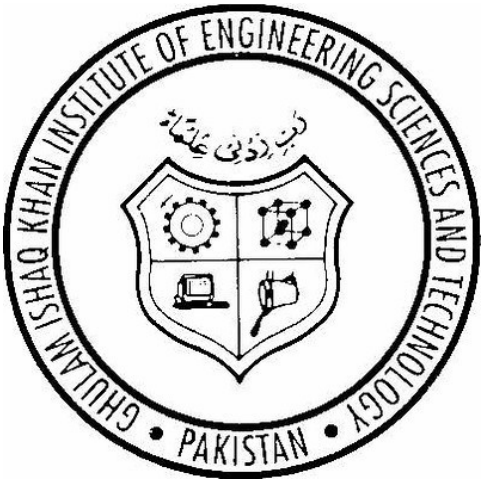
Semester Project Report

Payroll Management System

CS-101 Computing and AI



Submitted To:

Engr. Usama Manzoor

Submitted By:

	Student Name	**Student Reg #**	**Student Degree**
Student-1	Shayan Rizwan	2024585	BSCE
Student-2	Daniyal Ahmed	2024146	BSCE

Criteria	**Weightage**	**Student-1**	**Student-2**
Correctness and Completeness	30%		
Code Quality	20%		
Creativity and Practical Application	15%		
Report	10%		
Proposal	5%		
Viva	20%		

Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi

(You are required to follow the mentioned font size and style for the documentation. In this document heading are of 14 font size while regular text is of 12 font size, while the font style is "Times New Roman")

1. Main Features

Our Payroll Management System encompasses the following key features:

1. **Add** New Employee(s) – Simulation of a database
2. **Edit** Employee Information
3. **Generate** Employee Information
4. **View** Employee Personal Information
5. View Employee Payslip
6. View Monthly wise employee details

2. Types of Users & Requirements

A payroll management system with role-based access for admins to manage payroll and employees to view personal details and payslips.

2.1 Admin (Payroll System Manager)

2.1.1. Will be able to login into their position as ADMIN
2.1.2. Will be able to add a new employee into the program
2.1.3. Will be able to add diverse details related to the employee
2.1.4. Will be able to edit the details
2.1.5. Will be able to calculate and deduct applicable taxes like income tax
2.1.6. Will be able to compute the payslip of employee(s)
2.1.7 Will be able to print the employee details of chosen month

2.2 Employee (User Of Payroll System)

2.2.1 Will be able to view Personal information of the employee (in professionally aesthetic format)
2.2.2 Will be able to view Payslip of the employee (in professionally aesthetic format)

3. Requirements Breakdown

Below mentioned structure provides a clear and organized breakdown of the requirements for the payroll management system.

3.1 Role Selection (Admin/Employee):

3.1.1. Display the front screen with the option to select Admin or Employee
3.1.2. Validate the input from the user to check whether it is in the correct format or not (e.g. to check whether the input is an integer or not, and to output an error message if the input is incorrect)

3.2 Diverse Admin Functionalities:

3.2.1. Provide input fields for Employee ID, Name, Position, Monthly Salary, Date Of Birth, Address & SAVE the details in memory (using arrays) for use during the program session;
3.2.2 Edit employee information by prompting Admin to input the Employee ID to locate the record and allowing updating of required fields

3.2.3 Generate payslip by computing the daily rate, accounting for the overtime hours, outputting the gross salary, calculating the deductions (which comprise of income tax, leaves, and insurance contribution), and displaying the NET salary

3.2.4 Provide an option for Admin to return to the main menu after completing tasks

3.3 Employee Functionalities

3.3.1 View Personal information by accessing the data from the user inputs, in the previous window, and displaying in a proper format

3.3.2 Allow employees to view their respective payslips

4. Features to Coding Matrix

(In the following table you will mention the following items for each feature, mention the items in each column for each feature of your application)

Sr #	Feature Name	Programming Concept Used	No. of Files Used by File Handling	No. of Arrays Created	No. Pointers / References Used or Created	No. of Variables & Objects Created	Functions Created	Line of Code Written
1	Opening Window	Loops, switch case, functions	-	-	-	1	1	28
2	Admin Menu	File handling, loops, Input/Output Stream Handling, arrays, switch case, if-statements	1	5	-	18	5	179
3	Editing Employee Details	Loops, switch case, array manipulation, standard input handling, Input/Output Stream Handling	-	5	-	6	1	80
4	Employee Menu	Switch case, Loops, Arrays, Manual program termination	-	-	-	3	2	78
5	View Details	File handling, switch case, Input/Output Stream Handling	7	-	-	3	1	122

5. Code

```
#include <iostream>
#include <cmath>
#include <fstream>
#include <limits>
using namespace std;

// String is to be used for all the text inputs so that when, the user inputs, the spaces can be dealt with accordingly, using getline.
long long int EmployeeID;
string Name[100000];
string Position[100000];
string DOB[100000];
string address[100000];
int MonthSalary[100000];
string NewName[100000];
string NewPosition[100000];
string NewDOB[100000];
string Newaddress[100000];
float NewMonthSalary[100000]; // WHAT WILL HAPPEN IF I CHANGE THE MONTHSALARY FROM int TO float ?!
int DaysWorked;
int leaves;
float HourlyRate;
int Overtime;
float OvertimePay;
float DailyRate;
float GrossSalary;
char UserChoiceML; // This is simplified to as the following: the user choice for the medical leaves code block.
int ML; // ML = medical leaves;
float deduct;
float netsalary;
int Logout;
int ViewChoice;


void MainFunc();


void AdminMenu();


void EmployeeMenu();


bool validCheck;


void EditEmployee();
```

```
void intCheck();


void View();


int main () {

	cout << endl;
	cout << "\t\t\t ---------------------------------------------------- " << endl;
	cout << " \t\t\t\t Welcome To Our Payroll Management System! " << endl;
		cout << "\t\t\t ---------------------------------------------------- " << endl;
		cout << endl;


		MainFunc();


		return 0;

}



void MainFunc() {
		int MainChoice;
			while (true) {
			cout << "Please Select Your Role: " << endl;
	cout << "1. Admin " << endl;
	cout << "2. Employee " << endl;
	wcout << "ENTER YOUR CHOICE (1 OR 2) =  ";
	cin >> MainChoice;
	intCheck();
	switch (MainChoice) {
		case 1:
			AdminMenu();
			break;
		case 2:
			EmployeeMenu();
			break;
		default:
			cout << " --------- " << endl;
			continue;
	}


	break;


		}
```

```
}


void AdminMenu() {

	ofstream file("december.txt", ios::app);

	cout << endl;
	cout << endl;
	int AdminChoice;


	while (true) {
			cout << " ----------- " << endl;
	cout << " ADMIN MENU " << endl;
	cout << " ----------- " << endl;
			cout << "1. Add New Employee " << endl;
	cout << "2. Edit Employee Info " << endl;
	cout << "3. Generate Payslip " << endl;
	cout << "4. View Employee Details " << endl;
	cout << "5. PROGRAM TERMINATION " << endl;
	cout << "Enter Your Choice = ";
	cin >> AdminChoice;
	switch (AdminChoice) {
		case 1:
			cout << endl;
			cout << "Enter Employee ID    : ";
			cin >> EmployeeID;
			intCheck();
			cin.ignore();
			cout << "Enter Name           : ";
			getline(cin, Name[EmployeeID]);
			cout << "Enter D.O.B          : ";
			getline(cin, DOB[EmployeeID]);
			cout << "Enter Position       : ";
			getline(cin, Position[EmployeeID]);
			cout << "Enter Monthly Salary : ";
			cin >> MonthSalary[EmployeeID];
			intCheck();
			cin.ignore();
			cout << "Enter Address        : ";
			getline(cin, address[EmployeeID]);
			cout << endl << endl;

			file << EmployeeID << ", " << Name[EmployeeID] << ", " << DOB[EmployeeID]
<< ", " << Position[EmployeeID] << ", " << MonthSalary[EmployeeID] << ", " << address[EmployeeID]
<< endl;
			file.close() ;
			cout << "EMPLOYEE DETAILS ADDED SUCCESSFULLY! " << endl << endl;
```

```
                    break;

                case 2:
                    cout << endl;
                    cout << "Enter Employee ID To Edit : ";
                    cin >> EmployeeID;
                    cout << " ------------------- " << endl;
                    cout << " Current Information " << endl;
                    cout << " ------------------- " << endl;
                    cout << "NAME          : " << Name[EmployeeID] << endl;
                    cout << "D.O.B         : " << DOB[EmployeeID] << endl;
                    cout << "POSITION      : " << Position[EmployeeID] << endl;
                    cout << "MONTHLY SALARY : " << MonthSalary[EmployeeID] << endl;
                    cout << "ADDRESS : " << address[EmployeeID] << endl << endl;
                    cout << "What would you like to edit? " << endl;
                    EditEmployee();

                    break;

                case 3:
                    cout << endl;
                    cout << " -------------------- " << endl;
                    cout << " Generating Payslips " << endl;
                    cout << " -------------------- " << endl;
                    cout << "Enter Employee ID : ";
                    cin >> EmployeeID;
                    cout << "NAME            : " << Name[EmployeeID] << endl;
                    cout << "DESIGNATION     : " << Position[EmployeeID] << endl;
                    cout << "MONTHLY SALARY   : " << MonthSalary[EmployeeID] << endl <<
endl;
                    cout << "The Total Working Days = 22 " << endl << endl;
                    DailyRate = MonthSalary[EmployeeID]/22.00;
                    cout << "Daily Rate = " << DailyRate << endl;
                    cout << "Enter Days Worked = ";
                    cin >> DaysWorked;
                    if (DaysWorked <= 22&& DaysWorked >= 0) {
                        leaves = 22 - DaysWorked;
                        cout << "Enter the # of OVERTIME hours = ";
                        cin >> Overtime;
                        HourlyRate = DailyRate/9;
                        OvertimePay = Overtime*HourlyRate;
                        cout << endl << endl;
                        GrossSalary = (DailyRate*DaysWorked)+OvertimePay;
                        cout << " GROSS SALARY : " << GrossSalary << endl;
                        cout << " LEAVES : " << leaves << endl << endl;
                        cout << "Were there medical leaves (Y/N) ? ";
                        cin >> UserChoiceML;
                        if ((UserChoiceML == 'Y')||(UserChoiceML == 'y')) {
                            cout << "How many of them? ";
                            cin >> ML;
                            if (ML > 0 && ML <= 15) {
```

```
                                leaves = leaves - ML;
                                cout << "CASUAL LEAVES : " << leaves << endl;
                        }
                        else {
                                cout << "INVALID Input! " << endl;
                        }
                }
                else if (UserChoiceML == 'N'|| UserChoiceML == 'n') {
                        cout << "CASUAL LEAVES : " << leaves << endl;
                }
        }
        else {
                cout << "INVALID Input! There are 22 WORKING days. " << endl;
        }
        deduct = (0.05*GrossSalary)+(leaves*DailyRate)+1000;
        cout << "DEDUCTIONS  : " << deduct << endl << endl << endl;
        netsalary = GrossSalary - deduct;
        cout << "NET SALARY === " << netsalary << endl << endl;

        break;


case 4:

        View();

case 5:
        cout << endl;
        cout << " ---------------------------------------------------- " << endl;
                cout << "Would you like to: " << endl;
        cout << "1. Return To Main Menu " << endl;
        cout << "2. Logout " << endl;
        cout << "ENTER YOUR CHOICE = ";
        cin >> Logout;
        cout << endl << endl;
        switch (Logout) {
                case 1:

                        MainFunc();

                        break;

                case 2:
                cout << "\t\t\t\t ---------------------------------- " << endl;
                cout << "\t\t\t\t Thank You For Using Our Program! " << endl;
                cout << "\t\t\t\t ---------------------------------- " << endl;

        exit(0);

        break;
        }
```

```
	}


	} // This curly bracket corresponds to the while (true) loop.



}



void intCheck() {
	if (cin.fail()) {
		cin.clear();
		cin.ignore(1000, '\n');
		cout << "Invalid Input! Try Again. " << endl;
		validCheck = false;
	}
	else {
		validCheck = true;
	}
}


void EditEmployee() {
	int UserChoice;
	cout << "1. Name " << endl;
	cout << "2. D.O.B " << endl;
	cout << "3. Position " << endl;
	cout << "4. Monthly Salary " << endl;
	cout << "5. Address " << endl;
	cout << endl;
	cout << "ENTER YOUR CHOICE = ";
	cin >> UserChoice;
	switch (UserChoice) {
		case 1:
			cout << "Enter NEW Name : ";
			cin.ignore(); // This removes the input buffer from before, through the use of cin
operator, and prevents the compiler by going into the next line by way of '\n'.
			getline(cin, NewName[EmployeeID]);
			Name[EmployeeID] = NewName[EmployeeID];

			cout << endl << endl;


			cout << "EDITING DONE SUCCESSFULLY!" << endl << endl;

			break;
```

```
case 2:
        cout << "Enter NEW D.O.B : ";
        cin.ignore();
        getline(cin, NewDOB[EmployeeID]);
        DOB[EmployeeID] = NewDOB[EmployeeID];

        cout << endl << endl;


        cout << "EDITING DONE SUCCESSFULLY!" << endl << endl;


        break;

case 3:
        cout << "Enter NEW Position : ";
        cin.ignore();
        getline(cin, NewPosition[EmployeeID]);
        Position[EmployeeID] = NewPosition[EmployeeID];

        cout << endl << endl;


        cout << "EDITING DONE SUCCESSFULLY!" << endl << endl;


        break;

case 4:
        cout << "Enter NEW Monthly Salary : ";
        cin.ignore();
        cin >> NewMonthSalary[EmployeeID];
        MonthSalary[EmployeeID] = NewMonthSalary[EmployeeID];

        cout << endl << endl;


        cout << "EDITING DONE SUCCESSFULLY!" << endl << endl;


        break;

case 5:
        cout << "Enter NEW Address : ";
        cin.ignore();
        getline(cin, Newaddress[EmployeeID]);
        address[EmployeeID] = Newaddress[EmployeeID];

        cout << endl << endl;

        cout << "EDITING DONE SUCCESSFULLY!" << endl << endl;
```

```
                        break;
        }
}


void EmployeeMenu() {
        cout << endl;
        cout << endl;
        int EmployeeChoice;

        while (true) {
        cout << " --------------- " << endl;
        cout << " EMPLOYEE MENU " << endl;
        cout << " --------------- " << endl;
        cout << "1. View Personal Information " << endl;
        cout << "2. View Payslip " << endl;
        cout << "3. PROGRAM TERMINATION " << endl;
        cout << "Enter Your Choice = ";
        cin >> EmployeeChoice;
        switch (EmployeeChoice) {
                case 1:
                        cout << endl;
                        cout << "Kindly Enter Your Employee ID: ";
                        cin >> EmployeeID;
                        cout << " ---------------------------------------------- " << endl;
                        cout << "EMPLOYEE ID    : " << EmployeeID << endl;
                        cout << "NAME           : " << Name[EmployeeID] << endl;
                        cout << "DESIGNATION    : " << Position[EmployeeID] << endl;
                        cout << "D.O.B          : " << DOB[EmployeeID] << endl;
                        cout << "MONTHLY SALARY : " << MonthSalary[EmployeeID] << endl;
                        cout << "ADDRESS        : " << address[EmployeeID] << endl;
                        cout << " ---------------------------------------------- " << endl;

                        break;

                case 2:
                        cout << "Enter Your Employee ID : ";
                        cin >> EmployeeID;
                        cout << " ---------------------------------------------- " << endl;
                        cout << "EMPLOYEE ID       : " << EmployeeID << endl;
                        cout << "NAME              : " << Name[EmployeeID] << endl;
                        cout << "DESIGNATION       : " << Position[EmployeeID] << endl;
                        cout << "MONTHLY SALARY    : " << "$" << MonthSalary[EmployeeID] <<
endl;
                        cout << "DAILY RATE        : " << "$" << DailyRate << endl;
                        cout << "DAYS WORKED       : " << DaysWorked << "/22" << endl;
                        cout << endl;
                        cout << endl;
                        cout << "Gross Salary = " << "$" << GrossSalary << endl;
```

```
                cout << "Deductions   = " << "$" << deduct << endl;
                cout << "NET Salary   = " << "$" << netsalary << endl;
                cout << " ---------------------------------------------- " << endl;

                break;

        case 3:
                cout << "Would you like to: " << endl;
                cout << "1. Return To Main Menu " << endl;
                cout << "2. Logout " << endl;
                cout << "ENTER YOUR CHOICE = ";
                cin >> Logout;
                cout << endl << endl;
                switch (Logout) {
                        case 1:

                                MainFunc();

                                break;


                        case 2:
                                cout << "\t\t\t\t ---------------------------------- " << endl;
                                cout << "\t\t\t\t Thank You For Using Our Program! " << endl;
                                cout << "\t\t\t\t ---------------------------------- " << endl;

                                exit(0);

                                break;
                }



        }
        }
}


void View() {
        cout << "Please select the month for which you would like to view the employee data: ";
        cout << "1. June " << endl;
        cout << "2. July " << endl;
        cout << "3. August " << endl;
        cout << "4. September " << endl;
        cout << "5. October " << endl;
        cout << "6. November " << endl;
        cout << "7. December " << endl;
        cout << "ENTER YOUR CHOICE = ";
        cin >> ViewChoice;

        ifstream ViewFile;
```

```
   string text;
   // Variables declared once, to prevent re-declaration in each case.


         switch (ViewChoice) {
                  case 1:
         cout << "Displaying Data For June... " << endl;
         cout << endl;
         ViewFile.open("June.txt");
         if (ViewFile.is_open()) {
            while (getline(ViewFile, text)) { // getline allows for the whole lines to be output and displayed
line by line using endl.
               cout << text << endl;
            }
         } else {
            cout << "Error opening file for June." << endl;
         }
         ViewFile.close();  // It is necessary to close the file;
         break;

      case 2:
         cout << "Displaying Data For July... " << endl;
         cout << endl;
         ViewFile.open("July.txt");
         if (ViewFile.is_open()) {
            while (getline(ViewFile, text)) {
               cout << text << endl;
            }
         } else {
            cout << "Error opening file for July." << endl;
         }
         ViewFile.close();
         break;

      case 3:
         cout << "Displaying Data For August... " << endl;
         cout << endl;
         ViewFile.open("August.txt");
         if (ViewFile.is_open()) {
            while (getline(ViewFile, text)) {
               cout << text << endl;
            }
         } else {
            cout << "Error opening file for August." << endl;
         }
         ViewFile.close();
         break;

      case 4:
         cout << "Displaying Data For September... " << endl;
         cout << endl;
```

```
    ViewFile.open("September.txt");
    if (ViewFile.is_open()) {
        while (getline(ViewFile, text)) {
            cout << text << endl;
        }
    } else {
        cout << "Error opening file for September." << endl;
    }
    ViewFile.close();
    break;

case 5:
    cout << "Displaying Data For October... " << endl;
    cout << endl;
    ViewFile.open("October.txt");
    if (ViewFile.is_open()) {
        while (getline(ViewFile, text)) {
            cout << text << endl;
        }
    } else {
        cout << "Error opening file for October." << endl;
    }
    ViewFile.close();
    break;

case 6:
    cout << "Displaying Data For November... " << endl;
    cout << endl;
    ViewFile.open("November.txt");
    if (ViewFile.is_open()) {
        while (getline(ViewFile, text)) {
            cout << text << endl;
        }
    } else {
        cout << "Error opening file for November." << endl;
    }
    ViewFile.close();
    break;

case 7:
    cout << "Displaying Data For December... " << endl;
    cout << endl;
    ViewFile.open("December.txt");
    if (ViewFile.is_open()) {
        while (getline(ViewFile, text)) {
            cout << text << endl;
        }
    } else {
        cout << "Error opening file for December." << endl;
    }
    ViewFile.close();
```

```
        break;

    default:
        cout << "Invalid choice! Please enter a number between 1 and 7." << endl;
        break;
}


    }
```

6. Project Screenshots

```
                    ------------------------------------------------------
                            Welcome To Our Payroll Management System!
                    ------------------------------------------------------

Please Select Your Role:
1. Admin
2. Employee
ENTER YOUR CHOICE (1 OR 2) =  1
```

```
 -----------
 ADMIN MENU
 -----------
1. Add New Employee
2. Edit Employee Info
3. Generate Payslip
4. View Employee Details
5. PROGRAM TERMINATION
Enter Your Choice = 1

Enter Employee ID    : 12
Enter Name           : Shayan Rizwan
Enter D.O.B          : 17-02-2005
Enter Position       : Software Engineer
Enter Monthly Salary : 300000
Enter Address        : GIKI


EMPLOYEE DETAILS ADDED SUCCESSFULLY!
```

```
 -----------
 ADMIN MENU
 -----------
1. Add New Employee
2. Edit Employee Info
3. Generate Payslip
4. View Employee Details
5. PROGRAM TERMINATION
Enter Your Choice = 4
Please select the month for which you would like to view the employee data: 1. June
2. July
3. August
4. September
5. October
6. November
7. December
ENTER YOUR CHOICE = 1
Displaying Data For June...

101,John Doe,15-05-1990,Software Engineer,1234 Elm Street
102,Jane Smith,25-11-1985,Data Analyst,5678 Oak Avenue
103,Robert Brown,30-03-1988,Project Manager,9101 Pine Road
104,Emily White,22-07-1992,HR Specialist,1213 Maple Lane
105,Michael Johnson,05-08-1987,Software Developer,1415 Birch Blvd
106,Susan Davis,10-01-1984,UI/UX Designer,1617 Cedar Crescent
107,David Wilson,18-12-1991,Marketing Manager,1819 Elm Parkway
108,Linda Martinez,09-04-1986,Sales Executive,2021 Willow Way
109,William Garcia,03-09-1993,Content Writer,2223 Cherry Drive
110,Mary Rodriguez,11-06-1990,Product Owner,2425 Redwood Circle

 ------------------------------------------------
Would you like to:
1. Return To Main Menu
2. Logout
ENTER YOUR CHOICE =
```

```
 ---------------
 EMPLOYEE MENU
 ---------------
1. View Personal Information
2. View Payslip
3. PROGRAM TERMINATION
Enter Your Choice = 2
Enter Your Employee ID : 12
 ------------------------------------------------
EMPLOYEE ID       : 12
NAME              : Shayan Rizwan
DESIGNATION       : Software Engineer
MONTHLY SALARY    : $300000
DAILY RATE        : $13636.4
DAYS WORKED       : 18/22


Gross Salary = $248485
Deductions   = $40697
NET Salary   = $207788
 ------------------------------------------------
```

```
 -----------
 ADMIN MENU
 -----------
1. Add New Employee
2. Edit Employee Info
3. Generate Payslip
4. View Employee Details
5. PROGRAM TERMINATION
Enter Your Choice = 3

 --------------------
 Generating Payslips
 --------------------
Enter Employee ID : 12
NAME              : Shayan Rizwan
DESIGNATION       : Software Engineer
MONTHLY SALARY    : 300000

The Total Working Days = 22

Daily Rate = 13636.4
Enter Days Worked = 18
Enter the # of OVERTIME hours = 2


 GROSS SALARY : 248485
 LEAVES : 4

Were there medical leaves (Y/N) ? y
How many of them? 2
CASUAL LEAVES : 2
DEDUCTIONS  : 40697


NET SALARY === 207788
```