|  |
| --- |
|  |
| YSTEM |
|  |
|  |

**“EMPLOYEE MANAGEMENT SYSTEM”**

**MADE BY: aaNCHAL UPRETI (艾琳) 191764145**

EMPLOYEE MANAGEMENT SYSTEM

### ABOUT THE PROGRAM

The EMPLOYEE Management System is a program based on C++ made with tool DevC++. Employee Management system is an application that enables a company to create and store Employee Records. This application is helpful to department of the organization which maintains data of employees related to an organization.

USERS

This program can be operated by two kinds of users i.e.

* Admin: This user has access to all functionalities like add, modify, search, delete, display etc. of employee details.

Login id: admin

Password: password

* User: They are accessed only to display and search the employee

details.

Login id: user

Password: password

### FUNCTIONALITIES

Our Employee management system contains several functions which are as follow………

1. ADD A RECORD;

This function adds the details of the employee to the system. This is accessible only to admin.

1. Modify the existing Employee details;

This function helps to modify the employee details. This is accessible only to admin.

1. Searching Employee details with the help of Employee ID;

This function helps to search the details of the employee using employee id.

1. Delete the existing Employee details;

This function is also only accessible to admin. It lets the admin to delete the details of an employee from the system.

1. Displaying Employee details;

This function displays the list of employees with their details.

C++ Concepts

We have used several C++ concepts which classes are manipulators,functions, graphics, file handling.

**HEADER FILES: -**

A header file is a file with extension .h which contains C++ function declarations and macro definitions to be shared between several source files.

This project contains following Header Files:

1. #include<iostream> - To provide basic input/output for C++

2. #include<fstream> - Represent inpout/output file streams

3. #include<string> - To perform basic String operations

4. #include<iomanip> - To manipulate input/output

5. #include<conio.h> - To provide console input/output

6. #include<process.h> - Contains functions working with process

8. #include<windows.h> - To perform various built-in Windows function

**CLASSES: -**

Class is a way of binding data and its associated function together. It allows data (and function) to be hidden, if necessary, from external use. When defining class, we are creating new abstract data type that can be treated like any other built-in data type.

In this project, the concept of inheritance is used among two classes i.e.

* Class admin;

This class controls every operation that is performed in this project like adding data, displaying data, modifying data, searching data and deleting data. This is the parent class.

* Class user;

This class inherits information from the parent class i.e. class admin and is accessed to functions like search and display employee details only.

The usage of two classes allows the system to be encrypted and prevents important employee details to get misused by external source.

**MANIPULATORS: -**

Manipulators are operators that are used to format the data display. The most commonly used manipulator is endl.

The endl manipulator, when used in an output statement, causes a line feed to be inserted. It has the same effect as using the newline character “\n”.

In this project **User-Defined Manipulator** is also used.

Syntax of Defining a **User-Defined Manipulator**:

ostream & manipulator(ostream & ostr) //Outstream Manipulator  
 {  
 set of statements;  
 return ostr;   
 }

istream & manipulator(istream & istr) //Instream Manipulator  
{  
 set of statements;  
 return istr;  
}

The **User-Defined Manipulator** used in this project is:

ostream &nm(ostream &strname)

{

strname<<"\*WELCOME TO EMPLOYEE DATABASE MANAGEMENT SYSTEM \* ";

return strname;

}

**FUNCTIONS:-**

The basic work of function is to divide a program into parts. It basically states the structured programming. Another advantage of function is that it is possible to reduce the size of program by calling and using them at different places.

The Whole program is governed by main function. Calling of any function or returning the value is done by or in main function. Function call can be done by passing arguments or without passing arguments.

In this project following functions are used:

1. void choice() – This function checks the password entered by administrator and if the password is correct it allows to enter the database and also provides the user interface to choose the operations like end, decrypt, encrypt, write, delete, search, modify, read.

2. void login() – provides the user interface for administrator to enter the username and password.

3. void read() – used to call getdata() and decide that more record are to be entered or not.

4. void write() – used to create user interface for entering the data.

5. void modify() – to Modify Data

6. void modifydata() – to Modify Entries from Data

7. void getdata() – This function is called in read function to get the data of one employee.

8. void search() – to search a particular record in database.

9. void deleted() – to Delete Data

10. void display() – used to display the two lines(component of UI) after 1.5 sec.

11. void displaydata() – to Display Each Employees Data

13. void displayall() – is used to display the whole database together.

14. void gotoxy(short,short) – To specify x-axis and y-axis (gotoxy)

15. int getempid() – to Return Employee ID

**FILE HANDLING:-**

So far, we have been using the iostream standard library, which provides cin and cout methods for reading from standard input and writing to standard output respectively. This tutorial will teach you how to read and write from a file. This requires another standard C++ library called fstream, which defines three new data types:

A file must be opened before you can read from it or write to it. Either the ofstream or fstream object may be used to open a file for writing or 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.

void open(const char \*filename, ios::openmode);

Here, the first argument specifies the name and location of the file to be opened and the second0 argument of the open() member function defines the mode in which the file should be opened.

|  |  |
| --- | --- |
| **Data Type** | **Description** |
| Ofstream | This data type represents the output file stream and is used to create files and to write information to files. |
| Ifstream | This data type represents the input file stream and is used to read information from files. |
| Fstream | This data type represents the file stream generally, and has the capabilities of both ofstream and ifstream which means it can create files, write information to files, and read information from files. |

|  |  |
| --- | --- |
| **Mode Flag** | **Description** |
| ios::app | Append mode. All output to that file to be appended to the end. |
| ios::in | Open a file for reading. |
| ios::out | Open a file for writing. |
| Ios::binary | Opens in binary mode. |

Following File Handling Syntax are used in this project:

1. fstream f – To create file pointer named f

2. f.open("input.dat",ios::app | ios::out | ios::binary | ios::in) – To Open file input.dat.

3. f.write((char\*)&c,sizeof(c)) – To Write contents of object c to file

4. f.read((char\*)&c,sizeof(c)) – To Read from File

5. f.tellg() – Tells the current position pointer in file

6. f.seekg(0) – Sets the pointer of file to specified value in braces

**Graphics:-**

We have added text colour to several statements in our program as well as background colour to different screens.

To change the background and text colour, syntax is:

System(“COLOR A0”);

Color Codes are:

0 = Black  
1 = Blue  
2 = Green  
3 = Aqua  
4 = Red  
5 = Purple  
6 = Yellow  
7 = White  
8 = Grey  
9 = Light Blue  
A = Light Green  
B = Light Aqua  
C = Light Red  
D = Light Purple  
E = Light Yellow  
F = Bright White

EMPLOYEE MANAGEMENT SYSTEM

### USER MANUAL

LOGIN

After the welcome screen is displayed, in the login screen, enter the login id according to the type of user, then press enter key, Input the correct password and press enter again. After the successful login message, press enter again to go to main menu.

ADD A RECORD

Input ‘1’ and press enter to select “ADD A RECORD”. Now, enter the employee id and the system will check if the entered employee id exists previously or not. If the employee id exists already, then press enter key and input “Y” to choose to add another employee id. Now, input all the required employee details accordingly.

MODIFY A RECORD

Input ‘2’ and press enter to select “MODIFY A RECORD”. Now, input the employee id to be modified. Then, input the new employee details. Now, press enter to go to main menu.

SEARCH A RECORD

Input ‘3’ and press enter to select “SEARCH A RECORD”. Now, input the employee id to be searched. If found, then the required employee detail is displayed on the console.

DELETE A RECORD

Input ‘4’ and press enter to select “DELETE A RECORD”. Then, input the employee id whose details is to be deleted. Now, confirm the deletion of the record by entering ‘y’.

DISPLAYS ALL RECORD

Input ‘5’ and press enter to select “DISPLAYS ALL RECORD”. Now, the system will display all the existing records.

EXIT

Input ‘0’ to exit from the system.

