Project Report

EMPLOYEE MANAGEMENT SYSTEM

👨‍💻 Project Members:

Name: Kanishk Nagpal

Roll. No. : CO21328

Name: Krish Kathuria

Roll. No. : CO21337

Submitted to:

Dr. Sarabjeet Singh

📋Abstract

This report outlines the design and development of a computer software system to manage the employees of a company. The program was written in C++ programming language.

The design and ensuing program are modular in nature and make maximum use of abstract data types and of code re-useability. Particular attention is paid to the concepts of object-oriented programming.

The report includes the problem definition with an outline solution, scope for future development, conclusion, as well as the whole of the code that was written.

📃Contents

Problem Definition :

It is a brief overview of the status quo, and difficulties faced by the stakeholders. It also includes an approach that aims to solve real-world problems.

Outline Solution :

The working of the program concept is explained in step by step format. A flow of control between the objects and functions of the program is also explained in a comprehensive and detailed manner in the presentation submitted along the project.

Future Development :

A note on the scopes to which the project can be expanded. The modular nature of the project makes it easy to deconstruct and understand in order to add new elements and modify existing elements of the software so that is it more time efficient and ensures efficient memory utilization.

Conclusion :

A summary of the project covers all the major aspects, including the idea, intention, and functionality of the software. It also has a subsection that includes the user manual.

Source code :

The code is written in the C++ programming language.

💬Problem Definition

The structure of a cooperative enterprise is a hierarchy of founding members, board of directors, management, and employees. These broad levels of hierarchies contain various subcategories and expand horizontally according to the purpose they serve.

Management of data of every individual becomes a complex task as the enterprise expands and the number of individuals increases. A system that provides easy access is flexible and is able to sort and display data in a comprehensive yet concise manner is required to keep track of all employees and their records.

⚡Outline Solution

In order to solve the problem mentioned above we designed a software that:

- Builds the employee table
- Inserts new entries
- Deletes an entry
- Searches a record

🛸 Future Development

Although this implementation of the Employee Management System is on a relatively small scale with minimal yet necessary features, there is a scope for future developments. To start with, a GUI can be created using software like Visual Studio and enable a better user experience. This database is stored in MS Excel which is one of the leading DBMS software in the industry, which allows us easy manipulation of big data. It can be used as a database for websites like Employee Portals. Further, each record can be secured using unique security applications. This software is easily expandable just by making some precise changes in the source code.

📋User Manual

The setup process for first-time users is easy, just create an excel file with the following headings:

1. ID
2. First Name
3. Last Name
4. Gender
5. Age
6. Post
7. Salary

Then save it by the name **“Records”** as a **“.csv”** file.

This file created must be inside the same folder as the c++ program file.

Caution: Keep the Records.csv file closed while adding a record or deleting one.

🏁 Conclusion

The computer software system to manage the employees of a company which is written in C++ programming language was created to manage the data of every individual working at an enterprise that aims at providing easy-to-use software. The software makes an entry table in an excel sheet to which new entries can be added, deleted, and modified

according to the user's will. It is built with an object-oriented approach in order to make it flexible for future development.

💻Source Code

```
#include <iostream>
#include <conio.h>
#include <cmath>
#include <ctime>
#include <string>
#include <fstream>
#include <vector>
#include <windows.h>
#include <cctype>

using namespace std;

class Company
{
private:
    struct user
    {
        string fname, lname, post, department;
        int salary, ID, age;
        char gender;
    } user;

public:
    void choice();
    void getdata();
    void showdata();
    void deldata();
    void exitoption();

    string capitals(string);
};

int main()
{
    system("cls");
    system("COLOR f");

    cout << endl;
    cout << "\t============================" << endl;
    cout << "\t Employee Management System" << endl;
    cout << "\t============================";
    cout << endl;


    int counter();
    void sortdata();

    Company a;
    a.choice();

    sortdata();
    return 0;
}

void Company ::choice()
{
    int exittrig = 0;
    while (exittrig == 0)
    {
        char choice;
        cout << endl;
        cout << "What would you like to do: " << endl;
        cout << "1 -> Enter a new record." << endl;
        cout << "2 -> Show a particular record." << endl;
        cout << "3 -> Delete a particular record." << endl;
        cout << "4 -> Exit." << endl;
```

```
        cout << "\nEnter your choice: " << endl
             << "-> ";
        cin >> choice;
        Sleep(90);

        switch (choice)
        {
        case '1':
            Company ::getdata();
            break;

        case '2':
            Company ::showdata();
            break;

        case '3':
            Company ::deldata();
            break;

        case '4':
            exittrig = 1;
            exitoption();
            break;

        default:
            break;
        }
    }
}

void Company ::exitoption()
{
    system("cls");
    cout << "\t================================================" << endl;
    cout << "\t Thank you for using Employee Management System" << endl;
    cout << "\t================================================" << endl
         << endl;
}

int counter()
{

    ifstream countfile("Records.csv");
    string strtemp;
    int counter = 0;
    while (!countfile.eof())
    {
        getline(countfile, strtemp);
        counter++;
    }

    return counter - 1;
}

void Company ::getdata()
{

    system("cls");
    ifstream tempfile("Records.csv");
    int counter = 0, trigg = 0;
    char ch;
    string emptstr;
    if (tempfile.is_open())
    {

        while (!tempfile.eof())
        {
            getline(tempfile, emptstr);
            counter++;

            if (emptstr == "")
            {
                trigg = 1;
                break;
            }
        }
        tempfile.close();
    }
```

```
    if (trigg == 0)
    {
        counter = 0;
        ifstream countfile("Records.csv");
        while (!countfile.eof())
        {
            countfile >> ch;
            if (ch == ',')
            {
                counter++;
                cout << counter << endl;
            }
        }
        countfile.close();
    }
    else
    {
        counter--;
        counter = counter * 6;
    }

    system("cls");
    cout << "<<<<< Enter Data >>>>>" << endl
         << endl;
    cout << "Enter data for User " << (counter / 6) << ":" << endl;
    user.ID = (counter / 6);
    cout << "First Name: ";
    cin >> user.fname;
    cout << "Last Name: ";
    cin >> user.lname;
    cout << "Gender (M/F): ";
    cin >> user.gender;
    cout << "Age: ";
    cin >> user.age;
    cout << "Post: ";
    cin >> user.post;
    cout << "Salary: ";
    cin >> user.salary;

    fstream myFile;
    myFile.open("Records.csv", ios::app); // write
    if (myFile.is_open())
    {
        myFile << "\n"
               << user.ID << "," << capitals(user.fname) << "," << capitals(user.lname) << ","
               << (char)toupper(user.gender) << "," << user.age << "," << capitals(user.post) << "," << user.salary;
        myFile.close();
    }

    cout << endl
         << "<<<<< Record Entered Successfully >>>>>";
    Sleep(2000);
    system("cls");
}

void Company ::showdata()
{
    system("cls");
    int tempID;
    cout << "\t<<<<< Show Data >>>>>" << endl
         << endl;
    cout << "Enter ID of the User you want view: ";
    cin >> tempID;

    int max = counter();

    if (tempID <= max)
    {
        ifstream tempfile("Records.csv");

        string str;
        string head[7]{"ID", "First Name", "Last Name", "Gender", "Age", "Post", "Salary"};

        if (tempfile.is_open())
        {

            for (int i = 0; i <= tempID; i++)
            {
                getline(tempfile, str);
```

```
            if (i == tempID)
            {
                int len = str.length();
                int k = 0;
                cout << endl
                     << head[k] << ": ";

                for (int j = 0; j <= len; j++)
                {

                    if (str[j] == ',')
                    {
                        cout << endl;
                        k++;
                        cout << head[k] << ": ";
                    }

                    else
                    {
                        cout << str[j];
                    }
                }
            }
        }
            tempfile.close();
        }
    }

    else
    {
        cout << endl
             << "Data of User ID " << tempID << " doesn't exists." << endl;
    }

    cout << endl
         << "\t     <<<<< x >>>>>" << endl
         << endl;
}

void Company ::deldata()
{

    int tempID;
    system("cls");
    cout << "\t<<<<< Delete Data >>>>>" << endl
         << endl;
    cout << "Enter ID of the User you want to delete: ";
    cin >> tempID;

    int max = counter();
    vector<string> vec;
    string str;
    if (tempID <= max)
    {
        fstream tempfile("Records.csv", ios ::in);
        int i = 0;
        while (!tempfile.eof())
        {
            getline(tempfile, str);

            if (i == tempID)
            {
                str = "";
            }
            vec.push_back(str);
            i++;
        }
        tempfile.close();
        fstream tempfileo("Records.csv", ios ::out);

        int size = vec.size();

        for (int k = 0; k < size; k++)
        {
            tempfileo << vec.at(k) << endl;
        }
        tempfileo.close();
    }
```

```
    else
    {
        cout << endl
             << "Data of User ID " << tempID << " doesn't exists." << endl;
    }

    Sleep(100);
    cout << endl
         << "<<<<< Record Deleted Successfully >>>>>";
    Sleep(2000);
    system("cls");
}

void sortdata()
{

    vector<string> sortvec;
    string sortstr;
    fstream tempfilei("Records.csv", ios ::in);
    if (tempfilei.is_open())
    {
        while (!tempfilei.eof())
        {
            getline(tempfilei, sortstr);
            sortvec.push_back(sortstr);
        }
        tempfilei.close();
    }

    fstream tempfileo("Records.csv", ios ::out);
    if (tempfileo.is_open())
    {
        int p = 49; // 49 is the ASCII value of '1'.
        int len = sortvec.size();

        tempfileo << sortvec.at(0) << endl;
        for (int i = 1; i < len; i++)
        {

            for (int j = 0; j < len; j++)
            {
                if (sortvec.at(j)[0] == p)
                {
                    tempfileo << sortvec.at(j) << endl;
                    p++;
                }
            }
        }
        tempfileo.close();
    }
}

string Company::capitals(string text)
{

    for (int p = 0; p < text.length(); p++)
    {

        if (p == 0)
        {
            text[p] = toupper(text[p]);
        }

        else
        {
            text[p] = text[p];
        }
    }

    return text;
}
```