**SOURCE CODE**

#include <iostream>

#include <string>

#include <stdlib.h>

#include <conio.h>

#include <windows.h>

#include <fstream>

using namespace std;

struct Address

{

string house\_number;

string village;

string postoffice;

string city;

string state;

unsigned int PIN;

string country;

};

struct joining

{

int date\_join;

int month\_join;

int year\_join;

};

struct retirement

{

int date\_retire;

int month\_retire;

int year\_retire;

};

struct Node

{

struct Node \*next;

int Employee\_num;

string Employee\_name;

string Father\_name;

string department;

retirement emp\_retire;

joining emp\_join;

Address emp\_address;

};

Node \*start = NULL; //! saves the address of the first node

int count\_emp = 0;

void backup()

{

Node \*temp2 = start;

ofstream file;

file.open("Backup.csv");

if (!file)

{

cout << "Error in opening the file!!" << endl;

}

else

{

while (temp2 != NULL)

{

file

<< temp2->Employee\_num << "," << temp2->Employee\_name << "," << temp2- >Father\_name << "," << temp2->emp\_join.date\_join << "," << temp2- >emp\_join.month\_join << "," << temp2->emp\_join.year\_join << "," << temp2- >department << "," << temp2->emp\_retire.date\_retire << "," << temp2- >emp\_retire.month\_retire << "," << temp2->emp\_retire.year\_retire << "," << temp2- >emp\_address.house\_number << "," << temp2->emp\_address.village << "," << temp2- >emp\_address.postoffice << "," << temp2->emp\_address.city << "," << temp2- >emp\_address.state << "," << temp2->emp\_address.PIN << "," << temp2- >emp\_address.country << endl;

temp2 = temp2->next;

}

}

file.close();

}

int year\_input()

{

int input\_year;

bool check\_yearJoin = true;

do

{

cout << "Year : ";

cin >> input\_year;

if (input\_year >= 1947 && input\_year <= 2025)

{

check\_yearJoin = false;

}

else

{

cout << "Error in year!! Re-enter" << endl;

check\_yearJoin = true;

}

} while (check\_yearJoin);

return input\_year;

}

int month\_input()

{

int input\_month;

bool check\_monthJoin = true;

do

{

cout << "Month : ";

cin >> input\_month;

if (input\_month >= 1 && input\_month <= 12)

{

check\_monthJoin = false;

}

else

{

cout << "Error in month!! Re-enter" << endl;

check\_monthJoin = true;

}

} while (check\_monthJoin);

return input\_month;

}

int date\_input(Node \*ptr)

{

int input\_date;

bool check\_dateJoin = true;

do

{

cout << "Date : ";

cin >> input\_date;

if (ptr->emp\_join.month\_join >= 1 && ptr->emp\_join.month\_join <= 12)

{

if (ptr->emp\_join.month\_join == 1 || ptr->emp\_join.month\_join == 3 || ptr->emp\_join.month\_join == 5 || ptr->emp\_join.month\_join == 7 || ptr->emp\_join.month\_join == 8 || ptr->emp\_join.month\_join == 10 || ptr->emp\_join.month\_join == 12)

{

if (input\_date >= 1 && input\_date <= 31)

{

check\_dateJoin = false;

}

else

{

cout << "Error in date!! Re-enter" << endl;

check\_dateJoin = true;

}

}

else if (ptr->emp\_join.month\_join == 4 || ptr->emp\_join.month\_join == 6 || ptr- >emp\_join.month\_join == 9 || ptr->emp\_join.month\_join == 11)

{

if (input\_date >= 1 && input\_date <= 30)

{

check\_dateJoin = false;

}

else

{

cout << "Error in date!! Re-enter" << endl;

check\_dateJoin = true;

}

}

else if (ptr->emp\_join.month\_join == 2)

{

if (((ptr->emp\_join.year\_join % 4 == 0) && (ptr->emp\_join.year\_join % 100 != 0)) || ptr- >emp\_join.year\_join % 400 == 0)

{

if (input\_date >= 1 && input\_date <= 29)

{

check\_dateJoin = false;

}

else

{

cout << "Error in date!! Re-enter" << endl;

check\_dateJoin = true;

}

}

else

{

if (input\_date >= 1 && input\_date <= 28)

{

check\_dateJoin = false;

}

else

{

cout << "Error in date!! Re-enter" << endl;

check\_dateJoin = true;

}

}

}

}

} while (check\_dateJoin);

return input\_date;

}

void takeInput(int i)

{

Node \*ptr = new Node();

i++;

ptr->next = NULL;

ptr->Employee\_num = i;

cout << "Employee ID : " << ptr->Employee\_num << endl;

cout << "Employee name : ";

cin >> ptr->Employee\_name;

cin.ignore();

cout << "Father's name : ";

getline(cin, ptr->Father\_name);

bool check\_doj = true; //! can be removed

do

{

cout << "Enter joining info :-" << endl;

ptr->emp\_join.year\_join = year\_input();

ptr->emp\_join.month\_join = month\_input();

ptr->emp\_join.date\_join = date\_input(ptr);

check\_doj = false;

} while (check\_doj);

cin.ignore();

cout << "Department : ";

getline(cin, ptr->department);

bool dor\_check = true;

do

{

cout << "Enter retirement info :- " << endl;

ptr->emp\_retire.year\_retire = year\_input();

ptr->emp\_retire.month\_retire = month\_input();

ptr->emp\_retire.date\_retire = date\_input(ptr);

dor\_check = false;

} while (dor\_check);

cin.ignore();

cout << "Enter the Address:-" << endl;

cout << "House number : ";

getline(cin, ptr->emp\_address.house\_number);

cout << "Village/Town : ";

getline(cin, ptr->emp\_address.village);

cout << "Post Office : ";

getline(cin, ptr->emp\_address.postoffice);

cout << "City : ";

getline(cin, ptr->emp\_address.city);

cout << "State : ";

getline(cin, ptr->emp\_address.state);

bool check\_pin = true;

do

{

cout << "Pincode : ";

cin >> ptr->emp\_address.PIN;

if (ptr->emp\_address.PIN >= 100000 && ptr->emp\_address.PIN <= 999999)

{

check\_pin = false;

break;

}

else

{

cout << "Invalid PIN . Enter a valid PIN!!" << endl;

}

} while (check\_pin);

cin.ignore();

cout << "Country : ";

getline(cin, ptr->emp\_address.country);

if (start == NULL)

{

start = ptr;

// return start;

}

else

{

Node \*temp = start;

while (temp->next != NULL)

{

temp = temp->next;

}

temp->next = ptr;

}

// return start;

}

void printRecord()

{

Node \*print\_rec = start;

// int check = i;

// cout << print\_rec->Employee\_name;

if (print\_rec == NULL)

{

cout << "list is empty" << endl;

return;

}

cout << "The Entered records are:-" << endl;

while (print\_rec != NULL)

// while (print\_rec != NULL)

{

cout << "Details of Employee " << print\_rec->Employee\_num << " are:-" << endl;

cout << "Employee ID : " << print\_rec->Employee\_num << endl;

cout << "Employee name : " << print\_rec->Employee\_name << endl;

cout << "Father's name : " << print\_rec->Father\_name << endl;

// cout << "Joining info :-" << endl;

cout << "Joining : " << print\_rec->emp\_join.date\_join << "/" << print\_rec- >emp\_join.month\_join << "/" << print\_rec->emp\_join.year\_join << endl;

// cout << "Date of Joining : " << print\_rec->emp\_join.date\_join << endl;

// cout << "Month of Joining : " << print\_rec->emp\_join.month\_join << endl;

// cout << "Year of joining : " << print\_rec->emp\_join.year\_join << endl;

cout << "Department : " << print\_rec->department << endl;

cout << "Retirement : " << print\_rec->emp\_retire.date\_retire << "/" << print\_rec- >emp\_retire.month\_retire << "/" << print\_rec->emp\_retire.year\_retire << endl;

// cout << "Retirement info :-" << endl;

// cout << "Date of Retirement : " << print\_rec->emp\_retire.date\_retire << endl;

// cout << "Month of Retirement : " << print\_rec->emp\_retire.month\_retire << endl;

// cout << "Year of Retirement : " << print\_rec->emp\_retire.year\_retire << endl;

cout << "Address details :-" << endl;

cout << "House number : " << print\_rec->emp\_address.house\_number << endl;

cout << "Village/Town : " << print\_rec->emp\_address.village << endl;

cout << "Post office : " << print\_rec->emp\_address.postoffice << endl;

cout << "City : " << print\_rec->emp\_address.city << endl;

cout << "State : " << print\_rec->emp\_address.state << endl;

cout << "Pincode : " << print\_rec->emp\_address.PIN << endl;

cout << "Country : " << print\_rec->emp\_address.country << endl;

cout << endl;

print\_rec = print\_rec->next;

}

}

void print\_report(Node \*temp)

{

cout << "Details of Employee " << temp->Employee\_num << " are:-" << endl;

cout << "Employee ID : " << temp->Employee\_num << endl;

cout << "Employee name : " << temp->Employee\_name << endl;

cout << "Father's name : " << temp->Father\_name << endl;

cout << "Joining : " << temp->emp\_join.date\_join << "/" << temp->emp\_join.month\_join << "/" << temp->emp\_join.year\_join << endl;

cout << "Department : " << temp->department << endl;

cout << "Retirement : " << temp->emp\_retire.date\_retire << "/" << temp->emp\_retire.month\_retire << "/" << temp->emp\_retire.year\_retire << endl;

cout << "Address details :-" << endl;

cout << "House number : " << temp->emp\_address.house\_number << endl;

cout << "Village/Town : " << temp->emp\_address.village << endl;

cout << "Post office : " << temp->emp\_address.postoffice << endl;

cout << "City : " << temp->emp\_address.city << endl;

cout << "State : " << temp->emp\_address.state << endl;

cout << "Pincode : " << temp->emp\_address.PIN << endl;

cout << "Country : " << temp->emp\_address.country << endl;

cout << endl;

}

Node \*Search(Node \*start, int j)

{

Node \*ptr = start;

// string data1;

while (ptr != NULL)

{

if (ptr->Employee\_num == j)

{

// cout << "Employee Data Found..\n";

return ptr;

}

ptr = ptr->next;

}

return NULL;

}

void updateRecord(Node \*p)

{

bool flag = false;

bool loop\_run = true;

Node \*ptr1 = p;

if (ptr1 != NULL)

{

cout << endl;

cout << "1) Employee Name " << endl;

cout << "2) Father's Name " << endl;

cout << "3) Date of Joining " << endl;

cout << "4) Department " << endl;

cout << "5) Date of Retiremnet " << endl;

cout << "6) House Number " << endl;

cout << "7) Village " << endl;

cout << "8) Post Office " << endl;

cout << "9) City " << endl;

cout << "10) State " << endl;

cout << "11) PIN " << endl;

cout << "12) Country " << endl;

cout << "13) Exit " << endl;

int b;

upd:

cout << "What do you want to update :- ";

cin >> b;

cin.ignore();

switch (b)

{

case 1:

{

cout << "Enter Employee Name :- ";

getline(cin, ptr1->Employee\_name);

cout << "Record has been updated!!" << endl;

break;

}

case 2:

{

cout << "Enter Father's Name :- ";

getline(cin, ptr1->Father\_name);

cout << "Record has been updated!!" << endl;

break;

}

case 3:

{

cout << "Enter Joining info :- " << endl;

cout << "Enter Year:- ";

ptr1->emp\_join.year\_join = year\_input();

cout << "Enter Month:- ";

ptr1->emp\_join.month\_join = month\_input();

cout << "Enter Date:- ";

ptr1->emp\_join.date\_join = date\_input(ptr1);

cin.ignore();

cout << "Record has been updated!!" << endl;

break;

}

case 4:

{

cout << "Enter department:- ";

getline(cin, ptr1->department);

cout << "Record has been updated!!" << endl;

break;

}

case 5:

{

cout << "Enter Retirement info:- " << endl;

cout << "Enter Year: ";

ptr1->emp\_retire.year\_retire = year\_input();

cout << "Enter month: ";

ptr1->emp\_retire.month\_retire = month\_input();

cout << "Enter date: ";

ptr1->emp\_retire.date\_retire = date\_input(ptr1);

cin.ignore();

cout << "Record has been updated!!" << endl;

break;

}

case 6:

{

cout << "Enter House No. :- ";

getline(cin, ptr1->emp\_address.house\_number);

cout << "Record has been updated!!" << endl;

break;

}

case 7:

{

cout << "Enter Village :- ";

getline(cin, ptr1->emp\_address.village);

cout << "Record has been updated!!" << endl;

break;

}

case 8:

{

cout << "Enter Post Office :- ";

getline(cin, ptr1->emp\_address.postoffice);

cout << "Record has been updated!!" << endl;

break;

}

case 9:

{

cout << "Enter City :- ";

getline(cin, ptr1->emp\_address.city);

cout << "Record has been updated!!" << endl;

break;

}

case 10:

{

cout << "enter State :- ";

getline(cin, ptr1->emp\_address.state);

cout << "Record has been updated!!" << endl;

break;

}

case 11:

{

cout << "enter PIN:- ";

cin >> ptr1->emp\_address.PIN;

cin.ignore();

cout << "Record has been updated!!" << endl;

break;

}

case 12:

{

cout << "Enter country :- ";

getline(cin, ptr1->emp\_address.country);

cout << "Record has been updated!!" << endl;

break;

}

case 13:

{

break;

}

default:

{

cout << "Invalid input!! Re-enter choice" << endl;

goto upd;

}

}

}

else

{

cout << "Record does not exist" << endl;

}

}

Node \*delete\_Record(Node \*start, int i)

{

Node \*ptr\_delete = start;

Node \*temp = start;

Node \*prev = start;

int delete\_id;

// int count\_node;

bool flag\_delete = false;

bool search\_delete = true;

// Node \*temp = start;

if (ptr\_delete != NULL)

{

do

{

cout << "Enter the ID of the Employee you want to delete:- ";

cin >> delete\_id;

int count = 1;

do

{

if (delete\_id == ptr\_delete->Employee\_num)

{

flag\_delete = true;

break;

}

else

{

count++;

// Node \*temp = ptr\_delete;

ptr\_delete = ptr\_delete->next;

}

} while (ptr\_delete != NULL);

if (count == 1)

{

cout << "\nYOUR RECORD HAS BEEN DELETED" << endl;

return start->next;

}

while (temp->next != NULL && temp->Employee\_num != delete\_id)

{

prev = temp;

temp = temp->next;

}

if (temp->Employee\_num == delete\_id)

{

prev->next = temp->next;

delete (temp);

// createBackup();

cout << "\nYOUR RECORD HAS BEEN DELETED" << endl;

return start;

}

else

{

cout << "\n::NO RECORD EXISTS FOR EMPLOYEE " << delete\_id << "::" << endl;

;

break;

}

search\_delete = false;

} while (search\_delete);

count\_emp--;

}

else

{

cout << "List is empty" << endl;

return start;

}

}

void generate\_Report()

{

int choice;

// Node\*temp2 = start;

// Node\*temp3 = start;

string State, rep\_City, Country;

cout << "1)Employees with same state,city or country" << endl;

cout << "2)Employees with same joining date" << endl;

cout << "3)Employees with same retirement date" << endl;

cout << "4)Go to main menu" << endl;

// bool loop;

user\_choice:

cout << "Enter your choice : ";

cin >> choice;

switch (choice)

{

case 1:

{

Node \*temp = start;

if (temp == NULL)

{

cout << "List is empty!!" << endl;

return;

}

cout << "Enter City : ";

cin >> rep\_City;

cin.ignore();

cout << endl;

cout << "Enter State : ";

getline(cin, State);

cout << endl;

cout << "Enter Country : ";

getline(cin, Country);

bool check = false;

do

{

if (temp->emp\_address.city == rep\_City || temp->emp\_address.state == State || temp->emp\_address.country == Country)

{

// cout << "Records with same City :- " << endl;

print\_report(temp);

check = true;

temp = temp->next;

cout << endl

<< endl;

}

else

{

temp = temp->next;

}

} while (temp != NULL);

if (check = false)

{

cout << "No matching records!!" << endl;

}

}

break;

case 2:

{

Node \*temp = start;

int inp\_date, inp\_month, inp\_year;

cout << "Enter year : ";

inp\_year = year\_input();

cout << "Enter month : ";

inp\_month = month\_input();

cout << "Enter date : ";

inp\_date = date\_input(temp);

do

{

if (temp->emp\_join.date\_join == inp\_date && temp->emp\_join.month\_join == inp\_month && temp->emp\_join.year\_join == inp\_year)

{

cout << "Records with same Joining date are :- " << endl;

print\_report(temp);

temp = temp->next;

}

else

{

cout << "No matching Record!!" << endl;

}

} while (temp != NULL);

}

break;

case 3:

{

Node \*temp = start;

int inp\_date, inp\_month, inp\_year;

cout << "Enter year : ";

inp\_year = year\_input();

cout << "Enter month : ";

inp\_month = month\_input();

cout << "Enter date : ";

inp\_date = date\_input(temp);

do

{

if (temp->emp\_retire.date\_retire == inp\_date && temp->emp\_retire.month\_retire == inp\_month && temp->emp\_retire.year\_retire == inp\_year)

{

cout << "Records with same Retiring date are :- " << endl;

print\_report(temp);

temp = temp->next;

}

else

{

cout << "No matching Record!!" << endl;

}

} while (temp != NULL);

}

break;

case 4:

{

break;

}

default:

{

cout << "Invalid choice!! Re-enter" << endl;

goto user\_choice;

}

break;

}

}

void main\_display()

{

cout << "\t\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\t\t\t\t\t\*\t\t\t\t\t\t\t\t\t\t\t\t\t\*" << endl;

cout << "\t\t\t\t\t\*\t \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \t\*" << endl;

cout << "\t\t\t\t\t\*\t| \_\_\_| \\/ || \_\_\_ \\ | | \_ \\ \\ / / \_\_\_| \_\_\_| | \_\_\_ \\ \_\_\_/ \_\_ \\ \_ | \_\_\_ \\ \_ \\ \t\*" << endl;

cout << "\t\t\t\t\t\*\t| |\_\_ | . . || |\_/ / | | | | |\\ V /| |\_\_ | |\_\_ | |\_/ / |\_\_ | / \\/ | | | |\_/ / | | | \t\*" << endl;

cout << "\t\t\t\t\t\*\t| \_\_|| |\\/| || \_\_/| | | | | | \\ / | \_\_|| \_\_| | /| \_\_|| | | | | | /| | | | \t\*" << endl;

cout << "\t\t\t\t\t\*\t| |\_\_\_| | | || | | |\_\_\_\\ \\\_/ / | | | |\_\_\_| |\_\_\_ | |\\ \\| |\_\_\_| \\\_\_/\\ \\\_/ / |\\ \\| |/ / \t\*" << endl;

cout << "\t\t\t\t\t\*\t\\\_\_\_\_/\\\_| |\_/\\\_| \\\_\_\_\_\_/\\\_\_\_/ \\\_/ \\\_\_\_\_/\\\_\_\_\_/ \\\_| \\\_\\\_\_\_\_/ \\\_\_\_\_/\\\_\_\_/\\\_| \\\_|\_\_\_/ \t\*" << endl;

cout << "\t\t\t\t\t\*\t\t\t\t\t\t\t\t\t\t\t\t\t\*" << endl;

cout << "\t\t\t\t\t\*\t\t\t\t\t\t\t\t\t\t\t\t\t\*" << endl;

cout << "\t\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << endl

<< endl;

}

int main()

{

bool flag = true;

int i = 0;

do

{

system("cls");

main\_display();

cout << "\t\t\t\t\tEMPLOYEE MANAGEMENT SYSTEM" << endl;

cout << "\t\t\t\t\t1) Add record to the list" << endl;

cout << "\t\t\t\t\t2) Update a record " << endl;

cout << "\t\t\t\t\t3) Delete a record" << endl;

cout << "\t\t\t\t\t4) Print all the records" << endl;

cout << "\t\t\t\t\t5) Backup the records" << endl;

cout << "\t\t\t\t\t6) Report" << endl;

cout << "\t\t\t\t\t7) Exit" << endl;

cout << "\t\t\t\t\tYour choice :- ";

switch (getch())

{

case 49:

{

system("cls");

cout << "\n-----------------------------------------------" << endl;

cout << "\t Enter Employee's Data: " << endl;

cout << "-----------------------------------------------" << endl

<< endl;

takeInput(i);

i++;

count\_emp++;

cout << endl;

cout << "Successful Added!!!" << endl;

cout << "Press any key to go back to main menu" << endl;

getch();

}

break;

case 50:

{

system("cls");

int j;

update\_again:

cout << endl;

cout << "Enter the Employee ID you want to update : ";

cin >> j;

Node \*p = Search(start, j);

updateRecord(p);

char check;

cout << "Do you want to update another record? -> (Y/N) : ";

cin >> check;

check = toupper(check);

if (check == 'Y')

{

goto update\_again;

}

cout << "Press any key to go back to main menu" << endl;

getch();

}

break;

case 51:

{

system("cls");

start = delete\_Record(start, i);

count\_emp--;

cout << "Press any key to go back to main menu" << endl;

getch();

}

break;

case 52:

{

system("cls");

cout << "\n-----------------------------------------------" << endl;

cout << "\t Employee's Data: " << endl;

cout << "-----------------------------------------------" << endl

<< endl;

cout << "Total number of entered records : " << count\_emp << endl;

printRecord();

cout << "Press any key to go back to main menu" << endl;

getch();

break;

}

case 53:

{

system("cls");

backup();

cout << "data haas been backed up" << endl;

cout << "Press any key to go back to main menu" << endl;

getch();

}

break;

case 54:

{

system("cls");

generate\_Report();

cout << "Press any key to go back to main menu" << endl;

getch();

break;

}

case 55:

{

flag = false;

}

}

} while (flag);

}