EMS- A C++ based Employee Management System

Code:

#include <bits/stdc++.h>

using namespace std;

class Node

{

public:

Node \*prev;

Node \*next;

int empId;

string empName;

string department;

string position;

float salary;

public:

Node()

{

next = prev = NULL;

}

Node(int id, string name, string pos, string dept, float sal)

{

empId = id;

empName = name;

department = dept;

position = pos;

salary = sal;

next = NULL;

prev = NULL;

}

~Node()

{

cout << "Employee Deleted...." << endl;

}

};

class Employee

{

Node \*head;

Node \*tail;

public:

Employee()

{

head = NULL;

tail = NULL;

}

bool login();

void create();

void show();

void update(int id);

void remove(int id);

Node \*search(int id);

};

int main()

{

cout << " ~~~~~~ Welcome To Employee Management System ~~~~~~" << endl;

Employee emp;

int choice, id;

if (emp.login())

{

do

{

cout << "\nEmployee Management System\n";

cout << "1. Add Employee\n";

cout << "2. Display Employees\n";

cout << "3. Update Employee\n";

cout << "4. Remove Employee\n";

cout << "5. Search Employee\n";

cout << "6. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice)

{

case 1:

emp.create();

break;

case 2:

emp.show();

break;

case 3:

cout << "Enter Employee ID to update: ";

cin >> id;

emp.update(id);

break;

case 4:

cout << "Enter Employee ID to remove: ";

cin >> id;

emp.remove(id);

break;

case 5:

cout << "Enter Employee ID to search: ";

cin >> id;

if (emp.search(id))

{

cout << "Employee found!" << endl;

}

else

{

cout << "Employee not found!" << endl;

}

break;

case 6:

cout << "Exiting..." << endl;

break;

default:

cout << "Invalid choice!" << endl;

}

} while (choice != 6);

}

else

{

cout << "Incorrect ID or Password" << endl;

cout << "Try Again" << endl;

}

return 0;

}

bool Employee ::login()

{

int id;

string pass;

cout << "Please Enter ID and Password to Login" << endl;

cout << "Enter ID: ";

cin >> id;

cout << "Enter Password: ";

cin >> pass;

if (id == 207449 && pass == "soham1234")

{

cout << "Login Successfully!!!!" << endl;

return true;

}

return false;

}

void Employee::create()

{

int id;

string name, pos, dept;

float sal;

cout << "Enter Employee ID:";

cin >> id;

cout << "Employee Name: ";

cin >> name;

cout << "Employee Position: ";

cin >> pos;

cout << "Employee Department: ";

cin >> dept;

cout << "Employee Salary: ";

cin >> sal;

cout << "Employee Added Successfully!" << endl;

Node \*newNode = new Node(id, name, pos, dept, sal);

if (head == NULL)

{

head = newNode;

tail = newNode;

}

else

{

tail->next = newNode;

newNode->prev = tail;

tail = newNode;

}

}

void Employee ::show()

{

if (head == NULL)

{

cout << "No employees to display!" << endl;

return;

}

Node \*temp = head;

int i = 1;

while (temp != NULL)

{

cout << "Employee " << i++ << ":" << endl;

cout << "ID: " << temp->empId << endl;

cout << "Name: " << temp->empName << endl;

cout << "Position: " << temp->position << endl;

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

cout << "Salary: " << temp->salary << endl;

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

temp = temp->next;

}

}

void Employee ::update(int id)

{

Node \*temp = search(id);

if (temp == NULL)

{

cout << "Employee with ID:" << id << " ,Not Found!";

return;

}

cout << "Enter new details for Employee ID:" << id << ":" << endl;

cout << "Enter Name: ";

cin >> temp->empName;

cout << "Enter Position: ";

cin >> temp->position;

cout << "Enter Department: ";

cin >> temp->department;

cout << "Enter Salary: ";

cin >> temp->salary;

cout << "Employee details updated successfully!" << endl;

}

void Employee ::remove(int id)

{

Node \*emp = search(id);

if (emp == NULL)

{

cout << "Employee with ID " << id << " not found!" << endl;

return;

}

if (emp == head)

{

head = head->next;

if (head)

head->prev = NULL;

}

else if (emp == tail)

{

tail = tail->prev;

tail->next = NULL;

}

else

{

emp->prev->next = emp->next;

emp->next->prev = emp->prev;

}

delete emp;

cout << "Employee removed successfully!" << endl;

}

Node \*Employee::search(int id)

{

Node \*temp = head;

while (temp != NULL)

{

if (temp->empId == id)

{

return temp;

}

temp = temp->next;

}

return NULL;

}

**OUTPUT:**

~~~~~~ Welcome To Employee Management System ~~~~~~

Please Enter ID and Password to Login

Enter ID: 207449

Enter Password: soham1234

Login Successfully!!!!

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Remove Employee

5. Search Employee

6. Exit

Enter your choice: 1

Enter Employee ID:207464

Employee Name: Ritesh

Employee Position: ASE

Employee Department: PSU

Employee Salary: 500000

Employee Added Successfully!

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Remove Employee

5. Search Employee

6. Exit

Enter your choice: 1

Enter Employee ID:207445

Employee Name: Anish

Employee Position: ASE

Employee Department: PSU

Employee Salary: 50000.00

Employee Added Successfully!

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Remove Employee

5. Search Employee

6. Exit

Enter your choice: 2

Employee 1:

ID: 207464

Name: Ritesh

Position: ASE

Department: PSU

Salary: 500000

---------------------------------

Employee 2:

ID: 207445

Name: Anish

Position: ASE

Department: PSU

Salary: 50000

---------------------------------

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Remove Employee

5. Search Employee

6. Exit

Enter your choice: 3

Enter Employee ID to update: 207445

Enter new details for Employee ID:207445:

Enter Name: Anish Chauhan

Enter Position: Enter Department: PSU

Enter Salary: 50001

Employee details updated successfully!

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Remove Employee

5. Search Employee

6. Exit

Enter your choice: 4

Enter Employee ID to remove: 207464

Employee Deleted....

Employee removed successfully!

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Remove Employee

5. Search Employee

6. Exit

Enter your choice: 5

Enter Employee ID to search: 207445

Employee found!

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Remove Employee

5. Search Employee

6. Exit

Enter your choice: 6

Exiting...