

**Data Structure:**

**Smester Project:**

**Name:** Muhammad Aamir

Agha Shah Hussain

Abdullah Ahmed

**SAP ID:** 54676

53465

53639

**Semester:** 3rd

#include <iostream>

#include <string>

using namespace std;

struct Employee {

string name;

int id;

Employee\* next;

};

class EmployeeManagementSystem {

private:

Employee\* head;

int capacity;

int size;

string actionsLog;

public:

EmployeeManagementSystem(int capacity) {

head = nullptr;

this->capacity = capacity;

size = 0;

actionsLog = "";

}

~EmployeeManagementSystem() {

Employee\* current = head;

while (current != nullptr) {

Employee\* next = current->next;

delete current;

current = next;

}

}

void insertEmployee(const string& name, int id) {

if (size >= capacity) {

cout << "Cannot insert employee. System capacity reached." << endl;

return;

}

if (id < 10000 || id > 99999) {

cout << "ID must be a 5-digit number." << endl;

return;

}

Employee\* current = head;

while (current != nullptr) {

if (current->id == id) {

cout << "Employee with ID " << id << " already exists." << endl;

return;

}

current = current->next;

}

Employee\* newEmployee = new Employee{ name, id, head };

head = newEmployee;

size++;

actionsLog += "Employee added: " + name + " with ID " + to\_string(id) + "\n";

cout << "Employee " << name << " with ID " << id << " added." << endl;

}

void deleteEmployeeById(int id) {

Employee\* current = head;

Employee\* previous = nullptr;

while (current != nullptr) {

if (current->id == id) {

if (previous == nullptr) {

// Employee to delete is the head

head = current->next;

}

else {

previous->next = current->next;

}

delete current;

size--;

actionsLog += "Employee deleted with ID " + to\_string(id) + "\n";

cout << "Employee with ID " << id << " deleted." << endl;

return;

}

previous = current;

current = current->next;

}

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

}

void displayEmployees() {

Employee\* current = head;

cout << "Employees List (" << size << " employees):" << endl;

while (current != nullptr) {

cout << "Name: " << current->name << ", ID: " << current->id << endl;

current = current->next;

}

}

void displayOwnerInfo(int id) {

Employee\* current = head;

while (current != nullptr) {

if (current->id == id) {

cout << "Owner Information:" << endl;

cout << "Manager: Abdullah\nAssistant: Amir\nEmployee Leader: Agha" << endl;

return;

}

current = current->next;

}

cout << "Invalid ID. Access denied." << endl;

}

void viewImportantInformation(int id) {

if (id == 53639) {

cout << "Important Actions Log:" << endl;

cout << actionsLog << endl;

}

else {

cout << "Access denied. Only the manager can view this information." << endl;

}

}

};

int main() {

EmployeeManagementSystem ems(10);

int choice;

string name;

int id;

do {

cout << "Employee Management System Menu:\n";

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

cout << "2. Delete Employee by ID\n";

cout << "3. Display Employees\n";

cout << "4. View Owner Information (ID Required)\n";

cout << "5. View Important Information (Manager ID Required)\n";

cout << "6. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

cout << "Enter employee name: ";

cin >> name;

cout << "Enter employee ID (5-digit): ";

cin >> id;

ems.insertEmployee(name, id);

break;

case 2:

cout << "Enter employee ID to delete: ";

cin >> id;

ems.deleteEmployeeById(id);

break;

case 3:

ems.displayEmployees();

break;

case 4:

cout << "Enter your 5-digit ID to view owner information: ";

cin >> id;

ems.displayOwnerInfo(id);

break;

case 5:

cout << "Enter your manager ID to view important information: ";

cin >> id;

ems.viewImportantInformation(id);

break;

case 6:

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

break;

default:

cout << "Invalid choice. Please try again." << endl;

}

cout << endl;

} while (choice != 6);

return 0;

}