* **Code:**

#include <iostream>

#include <vector>

using namespace std;

class Employee {

public:

int id;

string name;

string department;

Employee(int id, string name, string department) {

this->id = id;

this->name = name;

this->department = department;

}

};

class Company {

public:

vector<Employee> employees;

void addEmployee(Employee employee) {

employees.push\_back(employee);

cout << "Employee added successfully." << endl;

}

void displayEmployee(int id) {

for (Employee employee : employees) {

if (employee.id == id) {

cout << "Employee ID: " << employee.id << endl;

cout << "Name: " << employee.name << endl;

cout << "Department: " << employee.department << endl;

return;

}

}

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

}

void deleteEmployee(int id) {

for (int i = 0; i < employees.size(); i++) {

if (employees[i].id == id) {

employees.erase(employees.begin() + i);

cout << "Employee deleted successfully." << endl;

return;

}

}

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

}

void updateEmployee(int id, string name, string department) {

for (Employee& employee : employees) {

if (employee.id == id) {

employee.name = name;

employee.department = department;

cout << "Employee information updated successfully." << endl;

return;

}

}

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

}

void displayAllEmployees() {

if (employees.empty()) {

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

} else {

for (Employee employee : employees) {

cout << "Employee ID: " << employee.id << endl;

cout << "Name: " << employee.name << endl;

cout << "Department: " << employee.department << endl;

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

}

}

}

};

int main() {

Company company;

company.addEmployee(Employee(1, "Edward", "Sales"));

company.addEmployee(Employee(2, "Sushant", "Engineering"));

company.addEmployee(Employee(3, "Alphonse", "Marketing"));

int choice;

do {

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

cout << "1. Add Employee" << endl;

cout << "2. Display Employee" << endl;

cout << "3. Delete Employee" << endl;

cout << "4. Update Employee" << endl;

cout << "5. Display All Employees" << endl;

cout << "0. Exit" << endl;

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1: {

int id;

string name, department;

cout << "Enter employee ID: ";

cin >> id;

cout << "Enter employee name: ";

cin.ignore();

getline(cin, name);

cout << "Enter employee department: ";

getline(cin, department);

company.addEmployee(Employee(id, name, department));

break;

}

case 2: {

int id;

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

cin >> id;

company.displayEmployee(id);

break;

}

case 3: {

int id;

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

cin >> id;

company.deleteEmployee(id);

break;

}

case 4: {

int id;

string name, department;

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

cin >> id;

cout << "Enter new name: ";

cin.ignore();

getline(cin, name);

cout << "Enter new department: ";

getline(cin, department);

company.updateEmployee(id, name, department);

break;

}

case 5: {

company.displayAllEmployees();

break;

}

case 0: {

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

break;

}

default: {

cout << "Invalid choice." << endl;

}

}

} while (choice != 0);

return 0;

}

* **Output:**



