#include <iostream>

#include <vector>

#include <iomanip>

using namespace std;

class Employee {

private:

string name;

int id;

double hoursWorked;

double hourlyRate;

public:

Employee(string n, int i, double h, double r)

: name(n), id(i), hoursWorked(h), hourlyRate(r) {}

int getID() { return id; }

string getName() { return name; }

double getHoursWorked() { return hoursWorked; }

double getHourlyRate() { return hourlyRate; }

void setHoursWorked(double h) { hoursWorked = h; }

void setHourlyRate(double r) { hourlyRate = r; }

double calculateSalary() {

return hoursWorked \* hourlyRate;

}

void displaySalarySlip() {

cout << "\n------ Salary Slip ------\n";

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

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

cout << "Hours Worked : " << hoursWorked << endl;

cout << "Hourly Rate : " << hourlyRate << endl;

cout << "Total Salary : " << calculateSalary() << endl;

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

}

};

class PayrollSystem {

private:

vector<Employee> employees;

public:

void addEmployee() {

string name;

int id;

double hours, rate;

cout << "Enter Employee ID: ";

cin >> id;

cout << "Enter Employee Name: ";

cin.ignore();

getline(cin, name);

cout << "Enter Hours Worked: ";

cin >> hours;

cout << "Enter Hourly Rate: ";

cin >> rate;

employees.push\_back(Employee(name, id, hours, rate));

cout << "Employee added successfully!\n";

}

void updateEmployee() {

int id;

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

cin >> id;

for (auto &emp : employees) {

if (emp.getID() == id) {

double hours, rate;

cout << "Enter new Hours Worked: ";

cin >> hours;

cout << "Enter new Hourly Rate: ";

cin >> rate;

emp.setHoursWorked(hours);

emp.setHourlyRate(rate);

cout << "Employee updated successfully!\n";

return;

}

}

cout << "Employee not found!\n";

}

void deleteEmployee() {

int id;

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

cin >> id;

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

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

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

cout << "Employee deleted successfully!\n";

return;

}

}

cout << "Employee not found!\n";

}

void searchEmployee() {

int id;

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

cin >> id;

for (auto &emp : employees) {

if (emp.getID() == id) {

emp.displaySalarySlip();

return;

}

}

cout << "Employee not found!\n";

}

void viewAllEmployees() {

if (employees.empty()) {

cout << "No employee records found!\n";

return;

}

cout << "\nList of Employees:\n";

cout << left << setw(10) << "ID" << setw(20) << "Name"

<< setw(15) << "Hours Worked" << setw(15)

<< "Hourly Rate" << setw(15) << "Salary" << endl;

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

for (auto &emp : employees) {

cout << left << setw(10) << emp.getID()

<< setw(20) << emp.getName()

<< setw(15) << emp.getHoursWorked()

<< setw(15) << emp.getHourlyRate()

<< setw(15) << emp.calculateSalary()

<< endl;

}

}

void generateSummaryReport() {

double totalPayroll = 0;

for (auto &emp : employees) {

totalPayroll += emp.calculateSalary();

}

cout << "\nTotal Payroll Amount: " << totalPayroll << endl;

}

};

int main() {

PayrollSystem system;

int choice;

do {

cout << "\n===== Employee Payroll System =====\n";

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

cout << "2. Update Employee\n";

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

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

cout << "5. View All Employees\n";

cout << "6. Generate Payroll Summary\n";

cout << "0. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1: system.addEmployee(); break;

case 2: system.updateEmployee(); break;

case 3: system.deleteEmployee(); break;

case 4: system.searchEmployee(); break;

case 5: system.viewAllEmployees(); break;

case 6: system.generateSummaryReport(); break;

case 0: cout << "Exiting system...\n"; break;

default: cout << "Invalid choice! Try again.\n";

}

} while (choice != 0);

return 0;

}