**Code:**

#include <iostream>

#include <string>

#include <vector>

using namespace std;

class Employee {

protected:

    string name;

    int age;

    double salary;

public:

    Employee(string n, int a, double s) : name(n), age(a), salary(s){}

    void display(){

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

        cout << "Age: " << age << endl;

        cout << "Salary: Rs." << salary << endl;

    }

    void updateEmployee(string newName, int newAge, double newSalary) {

        name = newName;

        age = newAge;

        salary = newSalary;

    }

    void updateEmployee(string newName) {

        name = newName;

    }

    void updateEmployee(int newAge) {

        age = newAge;

    }

    void updateEmployee(double newSalary) {

        salary = newSalary;

    }

};

class RegularEmployee : public Employee {

public:

    RegularEmployee(string n, int a, double s) : Employee(n, a, s) {}

    void display() {

        cout << "Regular Employee" << endl;

        Employee::display();

    }

};

class Manager : public Employee {

private:

    string dept;

public:

    Manager(string n, int a, double s, string d)

        : Employee(n, a, s), dept(d) {}

    void display(){

        cout << "Manager of " << dept << " Dept" << endl;

        Employee::display();

    }

    void updateDept(string newDept) {

        dept = newDept;

    }

};

class EmpManager {

private:

    vector<Employee> emps;

public:

    void addEmp(Employee emp) {

        emps.push\_back(emp);

    }

    void displayEmps()  {

        if (emps.empty()) {

            cout << "No employees available." << endl;

        }

        else{

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

                cout<<"\nEmployee "<<(i + 1)<<" Details:"<<endl;

                emps[i].display();

            }

        }

    }

    void updateEmp(int index, string newName, int newAge, double newSalary) {

        if (index >= 0 && index < emps.size()) {

            emps[index].updateEmployee(newName, newAge, newSalary);

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

        } else {

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

        }

    }

    void updateEmp(int index, string newName) {

        if (index >= 0 && index < emps.size()) {

            emps[index].updateEmployee(newName);

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

        } else {

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

        }

    }

    void updateEmp(int index, int newAge) {

        if (index >= 0 && index < emps.size()) {

            emps[index].updateEmployee(newAge);

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

        } else {

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

        }

    }

    void updateEmp(int index, double newSalary) {

        if (index >= 0 && index < emps.size()) {

            emps[index].updateEmployee(newSalary);

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

        } else {

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

        }

    }

    void deleteEmp(int index) {

        if (index >= 0 && index < emps.size()) {

            emps.erase(emps.begin() + index);

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

        } else {

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

        }

    }

    int getEmpsSize(){

        return emps.size();

    }

};

RegularEmployee createRegEmp() {

    string name;

    int age;

    double salary;

    cout << "Enter name: ";

    cin.ignore();

    getline(cin, name);

    cout << "Enter age: ";

    cin >> age;

    cout << "Enter salary: ";

    cin >> salary;

    return RegularEmployee(name, age, salary);

}

Manager createManager() {

    string name;

    int age;

    double salary;

    string dept;

    cout << "Enter name: ";

    cin.ignore();

    getline(cin, name);

    cout << "Enter age: ";

    cin >> age;

    cout << "Enter salary: ";

    cin >> salary;

    cout << "Enter department: ";

    cin >> dept;

    return Manager(name, age, salary, dept);

}

bool login() {

    string uname="anish";

    string pwd="anish123";

    string enteredUname, enteredPwd;

    cout << "Enter username: ";

    cin >> enteredUname;

    cout << "Enter password: ";

    cin >> enteredPwd;

    return enteredUname == uname && enteredPwd == pwd;

}

int main() {

    if (!login()) {

        cout << "Invalid username or password. Access denied!" << endl;

        return 0;

    }

    EmpManager mgr;

    int choice;

    while (true) {

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

        cout << "1. Add Employee\n";

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

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

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

        cout << "5. Exit\n";

        cout << "Enter your choice: ";

        cin >> choice;

        if (choice == 1) {

            int empType;

            cout << "Enter employee type (1 for Regular, 2 for Manager): ";

            cin >> empType;

            if (empType == 1) {

                RegularEmployee emp = createRegEmp();

                mgr.addEmp(emp);

                cout << "Regular employee added successfully!" << endl;

            } else if (empType == 2) {

                Manager emp = createManager();

                mgr.addEmp(emp);

                cout << "Manager added successfully!" << endl;

            } else {

                cout << "Invalid employee type!" << endl;

            }

        } else if (choice == 2) {

            mgr.displayEmps();

        } else if (choice == 3) {

            int index;

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

            cin >> index;

            index--;

            if (index >= 0 && index < mgr.getEmpsSize()) {

                int updateChoice;

                cout << "Choose what to update: \n";

                cout << "1. Name\n";

                cout << "2. Age\n";

                cout << "3. Salary\n";

                cout << "Enter your choice: ";

                cin >> updateChoice;

                if (updateChoice == 1) {

                    string newName;

                    cout << "Enter new name: ";

                    cin.ignore();

                    getline(cin, newName);

                    mgr.updateEmp(index, newName);

                } else if (updateChoice == 2) {

                    int newAge;

                    cout << "Enter new age: ";

                    cin >> newAge;

                    mgr.updateEmp(index, newAge);

                } else if (updateChoice == 3) {

                    double newSalary;

                    cout << "Enter new salary: ";

                    cin >> newSalary;

                    mgr.updateEmp(index, newSalary);

                } else {

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

                }

            }

            else{

                cout<<"Enter correct employee index!!"<<endl;

            }

        } else if (choice == 4) {

            int index;

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

            cin >> index;

            index--;

            mgr.deleteEmp(index);

        } else if (choice == 5) {

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

            break;

        } else {

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

        }

    }

    return 0;

}

**Output:**

Enter username: anish

Enter password: anish123

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Delete Employee

5. Exit

Enter your choice: 1

Enter employee type (1 for Regular, 2 for Manager): 1

Enter name: rahul

Enter age: 20

Enter salary: 20000

Regular employee added successfully!

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Delete Employee

5. Exit

Enter your choice: 1

Enter employee type (1 for Regular, 2 for Manager): 1

Enter name: sahil

Enter age: 23

Enter salary: 30000

Regular employee added successfully!

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Delete Employee

5. Exit

Enter your choice: 1

Enter employee type (1 for Regular, 2 for Manager): 2

Enter name: sohini

Enter age: 35

Enter salary: 60000

Enter department: devops

Manager added successfully!

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Delete Employee

5. Exit

Enter your choice: 1

Enter employee type (1 for Regular, 2 for Manager): 2

Enter name: varun

Enter age: 50

Enter salary: 100000

Enter department: testing

Manager added successfully!

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Delete Employee

5. Exit

Enter your choice: 2

Employee 1 Details:

Employee Name: rahul

Age: 20

Salary: Rs.20000

Employee 2 Details:

Employee Name: sahil

Age: 23

Salary: Rs.30000

Employee 3 Details:

Employee Name: sohini

Age: 35

Salary: Rs.60000

Employee 4 Details:

Employee Name: varun

Age: 50

Salary: Rs.100000

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Delete Employee

5. Exit

Enter your choice: 1

Enter employee type (1 for Regular, 2 for Manager): 1

Enter name: ram

Enter age: 31

Enter salary: 40000

Regular employee added successfully!

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Delete Employee

5. Exit

Enter your choice: 2

Employee 1 Details:

Employee Name: rahul

Age: 20

Salary: Rs.20000

Employee 2 Details:

Employee Name: sahil

Age: 23

Salary: Rs.30000

Employee 3 Details:

Employee Name: sohini

Age: 35

Salary: Rs.60000

Employee 4 Details:

Employee Name: varun

Age: 50

Salary: Rs.100000

Employee 5 Details:

Employee Name: ram

Age: 31

Salary: Rs.40000

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Delete Employee

5. Exit

Enter your choice: 3

Enter employee index to update: 4

Choose what to update:

1. Name

2. Age

3. Salary

Enter your choice: 2

Enter new age: 52

Employee age updated successfully.

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Delete Employee

5. Exit

Enter your choice: 2

Employee 1 Details:

Employee Name: rahul

Age: 20

Salary: Rs.20000

Employee 2 Details:

Employee Name: sahil

Age: 23

Salary: Rs.30000

Employee 3 Details:

Employee Name: sohini

Age: 35

Salary: Rs.60000

Employee 4 Details:

Employee Name: varun

Age: 52

Salary: Rs.100000

Employee 5 Details:

Employee Name: ram

Age: 31

Salary: Rs.40000

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Delete Employee

5. Exit

Enter your choice: 3

Enter employee index to update: 9

Enter correct employee index!!

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Delete Employee

5. Exit

Enter your choice: 4

Enter employee index to delete: 2

Employee deleted successfully.

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Delete Employee

5. Exit

Enter your choice: 2

Employee 1 Details:

Employee Name: rahul

Age: 20

Salary: Rs.20000

Employee 2 Details:

Employee Name: sohini

Age: 35

Salary: Rs.60000

Employee 3 Details:

Employee Name: varun

Age: 52

Salary: Rs.100000

Employee 4 Details:

Employee Name: ram

Age: 31

Salary: Rs.40000

Employee Management System

1. Add Employee

2. Display Employees

3. Update Employee

4. Delete Employee

5. Exit

Enter your choice: 5

Exiting program...