**Practical No:13**

**PROGRAM:**include <iostream>

#define max 20

using namespace std;

struct employee {

string name;

long int code;

string designation;

int exp;

int age;

};

int num;

employee emp[max];

void showMenu();

void build() {

cout << "\nBuild The Table\n";

cout << "Maximum Entries can be " << max << "\n";

cout << "Enter the number of entries required: ";

cin >> num;

if (num > max) {

cout << "Maximum number of entries is 20\n";

num = max;

}

for (int i = 0; i < num; i++) {

cout << "\nEntry " << i + 1 << ":\n";

cout << "Name: ";

cin >> emp[i].name;

cout << "Employee ID: ";

cin >> emp[i].code;

cout << "Designation: ";

cin >> emp[i].designation;

cout << "Experience (years): ";

cin >> emp[i].exp;

cout << "Age: ";

cin >> emp[i].age;

}

showMenu();

}

void insert() {

if (num < max) {

int i = num;

num++;

cout << "\nEnter the information of the new Employee:\n";

cout << "Name: ";

cin >> emp[i].name;

cout << "Employee ID: ";

cin >> emp[i].code;

cout << "Designation: ";

cin >> emp[i].designation;

cout << "Experience (years): ";

cin >> emp[i].exp;

cout << "Age: ";

cin >> emp[i].age;

} else {

cout << "\nEmployee Table Full\n";

}

showMenu();

}

void deleteIndex(int i) {

for (int j = i; j < num - 1; j++) {

emp[j] = emp[j + 1];

}

}

void deleteRecord() {

cout << "\nEnter the Employee ID to delete record: ";

int code;

cin >> code;

bool found = false;

for (int i = 0; i < num; i++) {

if (emp[i].code == code) {

deleteIndex(i);

num--;

found = true;

cout << "Record deleted successfully.\n";

break;

}

}

if (!found) {

cout << "Employee ID not found.\n";

}

showMenu();

}

void searchRecord() {

cout << "\nEnter the Employee ID to search: ";

int code;

cin >> code;

bool found = false;

for (int i = 0; i < num; i++) {

if (emp[i].code == code) {

cout << "\nRecord Found:\n";

cout << "Name: " << emp[i].name << "\n";

cout << "Employee ID: " << emp[i].code << "\n";

cout << "Designation: " << emp[i].designation << "\n";

cout << "Experience: " << emp[i].exp << " years\n";

cout << "Age: " << emp[i].age << "\n";

found = true;

break;

}

}

if (!found) {

cout << "Employee ID not found.\n";

}

showMenu();

}

void display() {

if (num == 0) {

cout << "\nNo records to display.\n";

} else {

cout << "\nEmployee Records:\n";

for (int i = 0; i < num; i++) {

cout << "\nRecord " << i + 1 << ":\n";

cout << "Name: " << emp[i].name << "\n";

cout << "Employee ID: " << emp[i].code << "\n";

cout << "Designation: " << emp[i].designation << "\n";

cout << "Experience: " << emp[i].exp << " years\n";

cout << "Age: " << emp[i].age << "\n";

}

}

showMenu();

}

void showMenu() {

int option;

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

cout << "Employee Management System\n";

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

cout << "\nAvailable Options:\n";

cout << "1. Build Table\n";

cout << "2. Insert New Entry\n";

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

cout << "4. Search a Record\n";

cout << "5. Display All Records\n"; // Moved before Exit

cout << "6. Exit\n";

cout << "Enter your choice: ";

cin >> option;

switch (option) {

case 1:

build();

break;

case 2:

insert();

break;

case 3:

deleteRecord();

break;

case 4:

searchRecord();

break;

case 5:

display();

break;

case 6:

cout << "Exiting Program.\n";

return;

default:

cout << "Invalid option. Please select between 1-6.\n";

showMenu();

}

}

int main() {

showMenu();

return 0;

}

**OUTPUT:**-------------------------

Employee Management System

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

Available Options:

1. Build Table

2. Insert New Entry

3. Delete Entry

4. Search a Record

5. Display All Records

6. Exit

Enter your choice: 1

Build The Table

Maximum Entries can be 20

Enter the number of entries required: 2

Entry 1:

Name: qwe

Employee ID: 123

Designation: hd

Experience (years): 10

Age: 21

Entry 2:

Name: tyu

Employee ID: 456

Designation: jail

Experience (years): 3

Age: 30

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

Employee Management System

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

Available Options:

1. Build Table

2. Insert New Entry

3. Delete Entry

4. Search a Record

5. Display All Records

6. Exit

Enter your choice: 2

Enter the information of the new Employee:

Name: poi

Employee ID: 678

Designation: kab

Experience (years): 4

Age: 30

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

Employee Management System

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

Available Options:

1. Build Table

2. Insert New Entry

3. Delete Entry

4. Search a Record

5. Display All Records

6. Exit

Enter your choice: 3

Enter the Employee ID to delete record: 123

Record deleted successfully.

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

Employee Management System

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

Available Options:

1. Build Table

2. Insert New Entry

3. Delete Entry

4. Search a Record

5. Display All Records

6. Exit

Enter your choice: 4

Enter the Employee ID to search: 456

Record Found:

Name: tyu

Employee ID: 456

Designation: jail

Experience: 3 years

Age: 31

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

Employee Management System

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

Available Options:

1. Build Table

2. Insert New Entry

3. Delete Entry

4. Search a Record

5. Display All Records

6. Exit

Enter your choice: 5

Employee Records

Record 1:

Name: tyu

Employee ID: 456

Designation: jail

Experience: 3 years

Age: 31

Record 2:

Name: poi

Employee ID: 678

Designation: kab

Experience: 4 years

Age: 30

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

Employee Management System

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

Available Options:

1. Build Table

2. Insert New Entry

3. Delete Entry

4. Search a Record

5. Display All Records

6. Exit

Enter your choice: 6

Exiting Program.