**EXPERIMENT – 16**

16. Develop a C program for implementing random access file for processing the employee

details.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define FILE\_NAME "employee.dat"

struct Employee {

int id;

char name[30];

float salary;

};

void addEmployee() {

FILE \*fp = fopen(FILE\_NAME, "ab");

if (!fp) {

perror("File open failed");

return;

}

struct Employee emp;

printf("Enter Employee ID: ");

scanf("%d", &emp.id);

printf("Enter Name: ");

scanf("%s", emp.name);

printf("Enter Salary: ");

scanf("%f", &emp.salary);

fwrite(&emp, sizeof(emp), 1, fp);

fclose(fp);

printf("Employee added successfully.\n");

}

void displayAll() {

FILE \*fp = fopen(FILE\_NAME, "rb");

if (!fp) {

perror("File open failed");

return;

}

struct Employee emp;

printf("\nEmployee Records:\n");

printf("----------------------------\n");

while (fread(&emp, sizeof(emp), 1, fp)) {

printf("ID: %d, Name: %s, Salary: %.2f\n", emp.id, emp.name, emp.salary);

}

fclose(fp);

}

void searchEmployee() {

int searchId;

printf("Enter Employee ID to search: ");

scanf("%d", &searchId);

FILE \*fp = fopen(FILE\_NAME, "rb");

if (!fp) {

perror("File open failed");

return;

}

struct Employee emp;

int found = 0;

while (fread(&emp, sizeof(emp), 1, fp)) {

if (emp.id == searchId) {

printf("Record found: ID: %d, Name: %s, Salary: %.2f\n", emp.id, emp.name, emp.salary);

found = 1;

break;

}

}

if (!found)

printf("Employee with ID %d not found.\n", searchId);

fclose(fp);

}

void updateEmployee() {

int updateId;

printf("Enter Employee ID to update: ");

scanf("%d", &updateId);

FILE \*fp = fopen(FILE\_NAME, "rb+");

if (!fp) {

perror("File open failed");

return;

}

struct Employee emp;

int found = 0;

while (fread(&emp, sizeof(emp), 1, fp)) {

if (emp.id == updateId) {

printf("Enter new Name: ");

scanf("%s", emp.name);

printf("Enter new Salary: ");

scanf("%f", &emp.salary);

fseek(fp, -sizeof(emp), SEEK\_CUR); // Move back to the record position

fwrite(&emp, sizeof(emp), 1, fp);

printf("Employee updated successfully.\n");

found = 1;

break;

}

}

if (!found)

printf("Employee with ID %d not found.\n", updateId);

fclose(fp);

}

void deleteEmployee() {

int deleteId;

printf("Enter Employee ID to delete: ");

scanf("%d", &deleteId);

FILE \*fp = fopen(FILE\_NAME, "rb");

FILE \*temp = fopen("temp.dat", "wb");

if (!fp || !temp) {

perror("File open failed");

return;

}

struct Employee emp;

int found = 0;

while (fread(&emp, sizeof(emp), 1, fp)) {

if (emp.id == deleteId) {

found = 1;

continue; // Skip writing this record to temp

}

fwrite(&emp, sizeof(emp), 1, temp);

}

fclose(fp);

fclose(temp);

remove(FILE\_NAME);

rename("temp.dat", FILE\_NAME);

if (found)

printf("Employee deleted successfully.\n");

else

printf("Employee with ID %d not found.\n", deleteId);

}

int main() {

int choice;

while (1) {

printf("\n--- Employee Management Menu ---\n");

printf("1. Add Employee\n");

printf("2. Display All Employees\n");

printf("3. Search Employee by ID\n");

printf("4. Update Employee by ID\n");

printf("5. Delete Employee by ID\n");

printf("6. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1: addEmployee(); break;

case 2: displayAll(); break;

case 3: searchEmployee(); break;

case 4: updateEmployee(); break;

case 5: deleteEmployee(); break;

case 6: exit(0);

default: printf("Invalid choice. Try again.\n");

}

}

return 0;

}

SAMPLE OUTPUT:

1. Add Employee

Enter ID: 101

Enter Name: Alice

Enter Salary: 50000

2. Display All Employees

ID: 101, Name: Alice, Salary: 50000.00

3. Update Employee by ID

Enter ID: 101

New Name: Alicia

New Salary: 55000