

Report On

Employee Management System

Submitted in partial fulfillment of the requirements of the Course project in
Semester III of Second Year Computer Engineering

By

Yash Kerkar (Roll No. S238267105)
Kunal Rajput (Roll No. S238307104)
Komal Sapatale (Roll No. S238317202)
Ankit Bari (Roll No. S238247105)

Supervisor

Prof. Soniya Khatu

Vidyavardhini's College of Engineering & Technology
Department of Artificial Intelligence And Data science



(2023-24)

Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence And Data science

CERTIFICATE

This is to certify that the project entitled “**Empolyee Management System**” is a bonafide work of "**Yash Kerkar** (Roll No. S238267105), **Kunal Rajput** (Roll No. S238307104), **Komal Sapatale** (Roll No. S238317202), **Ankit Bari** (Roll No. S238247105)" submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester III of Second Year Computer Engineering.

Supervisor

Prof. Soniya Khatu

Internal Examiner

External Examiner

Dr. Megha Trivedi

Head of Department

Dr. H.V. Vankudre

Principal

Contents

Title

Title page

Page Certificate

Content

Abstract

1. Problem statement

2. Block diagram

2.1: description And Working

3. ER diagram

4. Module Description

4.1: Employee Information Management

4.2: Employee Schedule Monitoring

4.3: Monitor Working Days And Holidays

4.4: Set Leave Processing

4.5: Generate Leave Reports

5. Software and Hardware Used and It's Programming

6. Code

7. Conclusion

8. References

Abstract

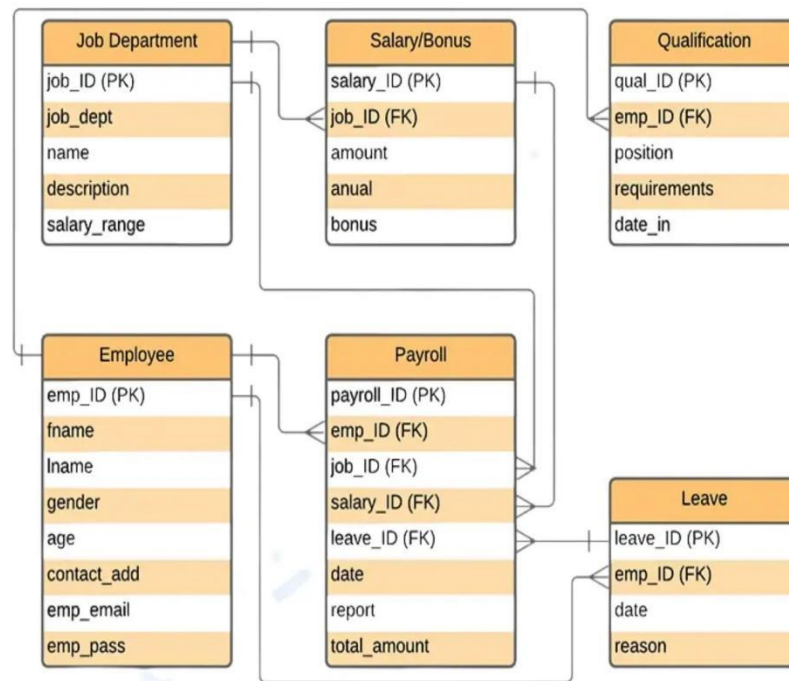
Employee Management System Project is an essentially software designed to keep track of employee information in any company. It stores data such as their employees' personal information. The goal of "Employee Management System" is to create a work center scheduling system. Scheduling is a technology that makes the process of informing activities and notifications in the company where it is implemented simple and even online.

The employee management system project gives managers a better idea of their employees and helps them plan and manage their work hours in order to cut costs and boost productivity. It gives appropriate directions and supervisions for employees. It also secures and manages information that are important to the employees including personal and work-related information.

Problem Statement

Develop an Employee Management System in C using a linked list to store and manage employee records efficiently. This system should allow users to add, display, search, edit, and delete employee data by their ID. The program should offer an interactive menu-driven interface for these operations, enhancing the organization's ability to manage employee information effectively and systematically. The primary objective is to create a user-friendly application for streamlined employee data management.

Block Diagram, Its Description and Working



• ER Diagram For Employee Management System (Database Deign)

ER diagram for an employee management system shows how the databases are connected to each other. It also shows how all the databases are logically related to each other. We can also create an ER diagram by drawing the figure of a different part of the employee management system and their properties and how they perform their task.

We can draw an ER diagram of the employee management system by drawing the design of the database. The sketch of the database became the storage of the database where the data comes and goes.

The employee management system ER Diagram is the model used to emphasize the system's database design. It gives deep understanding on employee management system database (back end) and its tables. This also guides in managing Employee database as well as the relationships of tables in it.

Modules Description

Here are the modules present in Employee Management System Project

Add Employee:

The "Add Employee" module is responsible for adding new employee records to the Employee Management System using a linked list data structure. It allows the user to input essential information about the employee, such as their name, ID, position, salary, and contact details. The module creates a new node in the linked list to represent the added employee and ensures that the linked list remains in a sorted order.

Display Employee:

The "Display Employee" module allows users to traverse and display the linked list containing all employee records. Users can access employee information in the order specified by the linked list's sorting criteria, making it easier to manage and access employee data within the system.

Search Employee:

The "Search Employee" module enables users to search for specific employees within the linked list-based system using various search criteria, such as name, ID, or position. It returns a list of matching employees, making it convenient to find and access information about employees based on specific search requirements.

Edit Employee:

The Edit Employee module in a C program using a linked list allows users to modify the details of an existing employee in an Employee Management System. This module provides the functionality to search for an employee by their unique ID, and if a match is found, it enables the user to update the employee's name, department, age, and salary. This is particularly useful for maintaining accurate and up-to-date records within the system.

Delete Employee:

The "Delete Employee" module allows authorized users to remove employee records from the linked list-based Employee Management System when needed. Users can search for the employee they wish to delete by providing the employee's unique ID or other identifying information. Once the employee is located, the module confirms the deletion action and removes the corresponding node from the linked list.

Software And Hardware Used and It's Programming

Hardware:

- **Intel Core i3 Processor:** The Intel Core i3 is a family of dual-core and quad-core processors known for their efficiency and balanced performance. It's commonly used in systems that require a good balance between processing power and power consumption. In your Employee Management System project, an Intel Core i3 processor would be sufficient for handling the software's requirements.
- **256GB Storage:** With 256GB of storage, you have a decent amount of space for storing your software, data files, and other project-related files. It provides ample storage for a project like an Employee Management System.
- **4GB of RAM:** Having 4GB of RAM (Random Access Memory) is suitable for a project like Employee Management System, allowing the software to run smoothly, perform data processing tasks, and handle user interactions effectively.

Software:

- **Operating System: Windows 10:** Windows 10 is a popular and widely used operating system developed by Microsoft. It provides a user-friendly interface, good software compatibility, and support for various programming languages. It's an excellent choice for developing and running your Employee Management System, as it offers a stable environment for your software.
- **Programming Language: C:** C is a versatile and widely used programming language known for its efficiency and portability. It's an excellent choice for developing system-level software like your Employee Management System. C allows for direct memory manipulation and provides a high degree of control over hardware, making it suitable for developing software that interacts with hardware components or databases.

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct Employee {
    int id;
    char name[50];
    char department[50];
    int age;
    float salary;
    struct Employee* next;
};
```

```
struct Employee* head = NULL;
```

```
void addEmployee(int id, char name[], char department[], int age, float salary) {
    struct Employee* newEmployee = (struct Employee*)malloc(sizeof(struct Employee));
    newEmployee->id = id;
    strcpy(newEmployee->name, name);
    strcpy(newEmployee->department, department);
    newEmployee->age = age;
    newEmployee->salary = salary;
    newEmployee->next = NULL;
```

```
    if (head == NULL) {
        head = newEmployee;
    } else {
        struct Employee* current = head;
        while (current->next != NULL) {
            current = current->next;
        }
        current->next = newEmployee;
    }
```

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

```
void displayEmployees() {
    struct Employee* current = head;
    if (current == NULL) {
        printf("No employee records found.\n");
        return;
    }
```

```
printf("\n%-5s%-20s%-15s%-5s%-10s\n", "ID", "Name", "Department", "Age",  
"Salary");  
printf("-----\n");
```

```
while (current != NULL) {  
printf("%-5d%-20s%-15s%-5d%-10.2f\n", current->id, current->name,  
current->department, current->age, current->salary);  
current = current->next;  
}  
}
```

```
void searchEmployee(int id) {  
struct Employee* current = head;  
while (current != NULL) {  
if (current->id == id) {  
printf("Employee found:\n");  
printf("ID: %d\nName: %s\nDepartment: %s\nAge: %d\nSalary: %.2f\n",  
current->id, current->name, current->department, current->age,  
current->salary);  
return;  
}  
current = current->next;  
}  
printf("Employee with ID %d not found.\n", id);  
}
```

```
void editEmployee(int id) {  
struct Employee* current = head;  
while (current != NULL) {  
if (current->id == id) {  
printf("Enter new name: ");  
scanf("%s", current->name);  
printf("Enter new department: ");  
scanf("%s", current->department);  
printf("Enter new age: ");  
scanf("%d", &current->age);  
printf("Enter new salary: ");  
scanf("%f", &current->salary);  
printf("Employee details updated successfully.\n");  
return;  
}  
current = current->next;  
}  
printf("Employee with ID %d not found.\n", id);  
}
```

```
void deleteEmployee(int id) {  
    struct Employee* current = head;  
    struct Employee* prev = NULL;
```

```
    while (current != NULL && current->id != id) {  
        prev = current;  
        current = current->next;  
    }
```

```
    if (current == NULL) {  
        printf("Employee with ID %d not found.\n", id);  
        return;  
    }  
    if (prev == NULL) {  
        head = current->next;  
    } else {  
        prev->next = current->next;  
    }  
    free(current);  
    printf("Employee with ID %d has been deleted.\n", id);  
}
```

```
int main() {  
    int choice, id, age;  
    char name[50];  
    char department[50];  
    float salary;
```

```
    while (1) {  
        printf("\nEmployee Management System\n");  
        printf("1. Add Employee\n");  
        printf("2. Display Employees\n");  
        printf("3. Search Employee\n");  
        printf("4. Edit Employee\n");  
        printf("5. Delete Employee\n");  
        printf("6. Exit\n");  
        printf("Enter your choice: ");  
        scanf("%d", &choice);
```

```
        switch (choice) {  
            case 1:  
                printf("Enter employee ID: ");  
                scanf("%d", &id);  
                printf("Enter employee name: ");  
                scanf("%s", name);  
                printf("Enter employee department: ");  
                scanf("%s", department);
```

```
printf("Enter employee age: ");
scanf("%d", &age);
printf("Enter employee salary: ");
scanf("%f", &salary);
addEmployee(id, name, department, age, salary);
break;
```

```
case 2:
displayEmployees();
break;
```

```
case 3:
printf("Enter the ID of the employee to search: ");
scanf("%d", &id);
searchEmployee(id);
break;
```

```
case 4:
printf("Enter the ID of the employee to edit: ");
scanf("%d", &id);
editEmployee(id);
break;
```

```
case 5:
printf("Enter the ID of the employee to delete: ");
scanf("%d", &id);
deleteEmployee(id);
break;
```

```
case 6:
exit(0);
```

```
default:
printf("Invalid choice. Please try again.\n");
}
}
```

```
return 0;
}
```

Output:

Add Employee:

```
Employee Management System
1. Add Employee
2. Display Employees
3. Search Employee
4. Edit Employee
5. Delete Employee
6. Exit
Enter your choice: 1
Enter employee ID: 10
Enter employee name: Yash
Enter employee department: Accounting
Enter employee age: 28
Enter employee salary: 25000
Employee added successfully.
```

```
Employee Management System
1. Add Employee
2. Display Employees
3. Search Employee
4. Edit Employee
5. Delete Employee
6. Exit
Enter your choice: 1
Enter employee ID: 20
Enter employee name: Ankit
Enter employee department: HR
Enter employee age: 25
Enter employee salary: 30000
Employee added successfully.
```

Display Employee:

Employee Management System

1. Add Employee
2. Display Employees
3. Search Employee
4. Edit Employee
5. Delete Employee
6. Exit

Enter your choice: 2

ID	Name	Department	Age	Salary
10	Yash	Accounting	28	25000.00
20	Ankit	HR	25	30000.00

Search Employee:

Employee Management System

1. Add Employee
2. Display Employees
3. Search Employee
4. Edit Employee
5. Delete Employee
6. Exit

Enter your choice: 3

Enter the ID of the employee to search: 20

Employee found:

ID: 20

Name: Ankit

Department: HR

Age: 25

Salary: 30000.00

Delete Employee:

Employee Management System

1. Add Employee
2. Display Employees
3. Search Employee
4. Edit Employee
5. Delete Employee
6. Exit

Enter your choice: 5

Enter the ID of the employee to delete: 10

Employee with ID 10 has been deleted.

Display Employee:

Employee Management System

1. Add Employee
2. Display Employees
3. Search Employee
4. Edit Employee
5. Delete Employee
6. Exit

Enter your choice: 2

ID	Name	Department	Age	Salary

20	Ankit	HR	25	30000.00

Conclusion:

The Employee Management System developed using linked lists in C demonstrates the power and flexibility of this data structure for organizing and managing employee data. Linked lists allow for dynamic addition, retrieval, modification, and removal of employee records, making it an efficient choice for such applications.

This project provides a solid foundation for understanding how linked lists work and how they can be applied in real-world scenarios. It showcases key functions for adding, displaying, searching, editing, and deleting employee records, offering a valuable learning experience for those interested in data structures and software development in C. While this project serves as an excellent starting point, there is ample room for expansion and enhancement to create a more comprehensive and robust employee management solution.

Thank You...

References:

- **Websites:-**

<https://lnu.diva-portal.org/smash/get/diva2:204828/FULLTEXT01.pdf>

<https://www.studocu.com/in/document/st-josephs-college-of-law-bangalore/java-project/employee-management-system-project-report/32274216employee-management> • [GitHub Topics](#)

<http://www.google.com>

<http://www.microsoft.com>

<http://www.codeproject.com>

<http://www.vb123.com>

<http://www.vbcode.com>

- **BOOKS:-**

Mastering Visual Basic 6(Paperback)

Visual Basic Black Book(Paperbac

Databas Development in Visual Basic

Teach Yourself Visual Basic 6 McGraw Hill

Software Engineering-Robert A. Pressman(McGraw Hill)