

# Data structures Project

## Employee record management system using linked list

**Problem:** Create an employee Record Management system using linked list that can perform the following operations:

- Insert employee record
- Delete employee record
- Update employee record
- Show details of an employee
- Search an employee
- Update the salary of an employee

The employee record should contain the following items

- Name of Employee
- ID of Employee
- First day of work
- Phone number of the employee
- Address of the employee
- Work hours
- Salary

**Extra: Design a Graphical user interface for the employee Record Management system**

### Approach:

With the basic knowledge of operations on Linked Lists like insertion, deletion of elements in the Linked list, the employee record management system can be created. Below are the functionalities explained that are to be implemented:

**Check Record:** It is a utility function of creating a record it checks before insertion that the Record Already Exist or not. It uses the concept of checking for a Node with given Data in a linked list.

**Create Record:** It is as simple as creating a new node in the Empty Linked list or inserting a new node in a non-Empty linked list.

**Smart Search Record:** Search a Record is similar to searching for a key in the linked list. Here in the employee record key is the ID number as a unique for every employee.

**Delete Record:** Delete Record is similar to deleting a key from a linked list. Here the key is the ID number. Delete record is an integer returning function it returns -1 if no such

record with a given roll number is found otherwise it deletes the node with the given key and returns 0.

Show Record: It shows the record is similar to printing all the elements of the Linked list.

Update salary: It add 2% of the salary for every extra hour. By default, 32 hours are required for every employee.

## Recommendations:

Although the implementation of exception handling is quite simple few things must be taken into consideration before designing such a system:

1. ID must be used as a **key** to distinguish between two different records, so while inserting record check whether this record already exists in our database or not, if it already exists then immediately report to the user an error message.