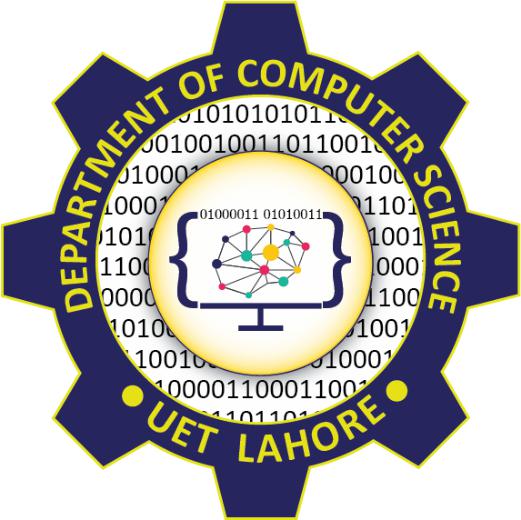
**DSA FINAL PROJECT PROPOSAL**



**Session 2023 - 2027**

**Submitted by:**  
Arwa Mahwash 2023-CS-16

**Submitted To:**

Sir Nazeef-Ul-Haq

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

Table of Contents

[Organizational Hierarchy Manager 3](#_Toc182818391)

[Project Description: 3](#_Toc182818392)

[Objectives 3](#_Toc182818393)

[Key Features 3](#_Toc182818394)

[Add/Remove Employees or Departments: 3](#_Toc182818395)

[Find Reporting Chains: 3](#_Toc182818396)

[List Subordinates: 3](#_Toc182818397)

[Dynamic Reorganization: 3](#_Toc182818398)

[Hierarchy Analysis: 4](#_Toc182818399)

[Search and Sort: 4](#_Toc182818400)

[Data Structures Used 4](#_Toc182818401)

[Tree: 4](#_Toc182818402)

[Graph: 4](#_Toc182818403)

[Stack: 4](#_Toc182818404)

[Queue: 4](#_Toc182818405)

[Linked List: 4](#_Toc182818406)

[Algorithms Used: 4](#_Toc182818407)

[Depth-First Search (DFS): 4](#_Toc182818408)

[Breadth-First Search (BFS): 4](#_Toc182818409)

[Conclusion 4](#_Toc182818410)

## Organizational Hierarchy Manager

## Project Description:

The Organizational Hierarchy Manager is a software application designed to manage and analyze the structure of an organization. It provides functionalities to add, remove, and manage employees and departments, find reporting chains, and dynamically reorganize the hierarchy. This project uses fundamental data structures such as graphs, trees, stacks, queues, and linked lists while integrating traversal algorithms like DFS and BFS to handle organizational data efficiently.

# Objectives

* Represent the hierarchical structure of an organization using trees and graphs.
* Implement features for managing and analyzing organizational data dynamically.
* Demonstrate practical applications of core data structures (graphs, trees, stacks, queues, linked lists).
* Showcase traversal algorithms (DFS, BFS) and sorting algorithms where applicable.

# Key Features

## Add/Remove Employees or Departments:

* + Dynamically update the hierarchy.
  + Ensure relationships between entities remain consistent.

## Find Reporting Chains:

* + Trace the reporting chain of an employee to the CEO using **stack**.

## List Subordinates:

* + Display all employees under a manager using **queue** for level-order traversal.

## Hierarchy Analysis:

* + Find the longest chain of command.
  + Count employees at each level of the hierarchy.

## Search and Sort:

* + Search employees by name or ID using DFS or BFS.
  + Sort employees by level, seniority, or name.

# Data Structures Used

## Tree:

* + Modelling the hierarchy (eg. CEO → Managers → Employees)

## Graph:

* + Represent cross-department collaborations or secondary reporting relationships.

## Stack:

* + Backtracking in reporting chain analysis or tracing changes in hierarchy.

## Queue:

* + Level-order traversal for listing employees.

## Linked List:

* + Manage dynamic lists of employees under a manager or departments under a parent.

# Algorithms Used:

## Depth-First Search (DFS):

* + Explore reporting chains.
  + Find specific employees or departments.

## Breadth-First Search (BFS):

* + Level-order traversal to list subordinates or count employees

# ****Conclusion****

The **Organizational Hierarchy Manager** effectively applies graphs, trees, and other core data structures like stacks, queues, and linked lists to manage and analyze hierarchical relationships in an organization.