# Name: Talha Rafiq

# Class: BSAI (2A)

# Roll No: 029 Employee Management System Project Report

## 1. Functional Operations

This project implements an Employee Management System that allows for the management of employees, including adding, updating, deleting, promoting, and displaying employee records. The program utilizes classes to represent different types of employees (Manager and Worker) and persists data using CSV files.

## 2. CRUD

The system implements full CRUD (Create, Read, Update, Delete) operations:  
- \*\*Create\*\*: Add new employees (Manager or Worker) to the system.  
- \*\*Read\*\*: Display all employee records with relevant details.  
- \*\*Update\*\*: Modify the details of an existing employee, including their name, age, salary, and specific attributes such as department or hours worked.  
- \*\*Delete\*\*: Remove an employee record from the system.

## 3. Scope

The Employee Management System is designed for small to medium-sized organizations to efficiently manage employee data. The program is console-based, making it suitable for deployment in environments where a simple interface suffices. The extensible design allows for additional employee types or attributes to be integrated with minimal changes to the core code.

## 4. Problem Statement

Managing employee records manually can be time-consuming and error-prone. Organizations often face challenges in keeping employee data up-to-date, calculating salary increments, and tracking specific employee attributes like hours worked or departmental assignments. The Employee Management System addresses these challenges by providing a structured, digital solution for maintaining accurate and accessible employee records.

## 5. Code Overview

The system is implemented in Python and comprises several key components:  
- \*\*Employee Class\*\*: A base class encapsulating shared attributes (name, age, salary) and methods.  
- \*\*Manager and Worker Classes\*\*: Derived classes representing specific types of employees with additional attributes.  
- \*\*CSV File Handling\*\*: Functions to load and save employee data to a CSV file for persistence.  
- \*\*User Interface\*\*: Console-based interaction allowing users to log in and manage employee data via a menu-driven system.

**Thank You**