# HR Management System (HRM) –

# Final Project Report

## Project Information

Project Title: HR Management System

Course: Basics of Python

Instructor: Sir Usama

Submission Date: 3-Oct-25

## Team Members

1. Muhammad Ali Ahmar – S2023266041

2. Muhammad Zaeem Sheikh – S2023266043

## Introduction

The HR Management System (HRM) project is designed to simulate basic HR operations such as employee management, leave tracking, payroll generation, and project tracking. The project provides an interactive menu-driven interface for users to manage employee records and projects effectively.

## Objectives

- To develop a simple console-based HR Management System.  
- To implement functionalities such as adding/removing employees, applying leave, and generating pay slips.  
- To practice the use of Python data structures like dictionaries and lists.  
- To enhance skills in modular programming and function-based design.

## System Features

1. Add Employee: Allows the user to add employee details like ID, name, position, and salary.  
2. Remove Employee: Enables removal of an employee from the system using their ID.  
3. Apply for Leave: Employees can apply for leave, and the system records leave days.  
4. Create Pay Slip: Generates a pay slip displaying employee details, salary, and leave information.  
5. Show Projects: Displays the list of ongoing projects.  
6. Display All Employees: Provides a list of all employees along with their details.  
7. Add Project: Adds a new project to the project list.  
8. Exit: Terminates the program.

## Code Explanation

The project is implemented in Python and makes use of fundamental programming concepts such as:  
- Dictionaries to store employee information.  
- Lists to maintain a record of projects.  
- Functions for modular implementation of tasks like adding employees, applying leaves, and generating pay slips.  
- A menu-driven loop to allow users to navigate through different options.

## CODE

employees = {}

projects = []

def add\_employee():

emp\_id = input("Enter Employee ID: ")

if emp\_id in employees:

print("Employee already exists!\n")

return

name = input("Enter Employee Name: ")

position = input("Enter Position: ")

salary = float(input("Enter Salary: "))

employees[emp\_id] = {

"name": name,

"position": position,

"salary": salary,

"leave\_days": 0

}

print(f"Employee {name} added successfully!\n")

def remove\_employee():

emp\_id = input("Enter Employee ID to remove: ")

if emp\_id in employees:

del employees[emp\_id]

print("Employee removed successfully!\n")

else:

print("Employee not found!\n")

def apply\_leave():

emp\_id = input("Enter Employee ID: ")

if emp\_id in employees:

days = int(input("Enter number of leave days: "))

employees[emp\_id]["leave\_days"] += days

print(f"Leave applied for {employees[emp\_id]['name']} ({days} days)\n")

else:

print("Employee not found!\n")

def create\_payslip():

emp\_id = input("Enter Employee ID: ")

if emp\_id in employees:

emp = employees[emp\_id]

print("\n--- Pay Slip ---")

print(f"Employee ID: {emp\_id}")

print(f"Name: {emp['name']}")

print(f"Position: {emp['position']}")

print(f"Salary: {emp['salary']} PKR")

print(f"Leave Days Taken: {emp['leave\_days']}")

print("----------------\n")

else:

print("Employee not found!\n")

def show\_projects():

if not projects:

print("No projects found!\n")

return

print("\n--- Ongoing Projects ---")

for i, proj in enumerate(projects, 1):

print(f"{i}. {proj}")

print("-------------------------\n")

def add\_project():

project\_name = input("Enter new project name: ")

if project\_name in projects:

print("This project already exists!\n")

else:

projects.append(project\_name)

print(f"Project '{project\_name}' added successfully!\n")

def display\_all\_employees():

if not employees:

print("No employees found!\n")

return

print("\n--- Employee List ---")

for emp\_id, emp in employees.items():

print(f"ID: {emp\_id} | Name: {emp['name']} | Position: {emp['position']} | "

f"Salary: {emp['salary']} PKR | Leave Days: {emp['leave\_days']}")

print("----------------------\n")

def menu():

while True:

print("====== HR Management System ======")

print("1. Add Employee")

print("2. Remove Employee")

print("3. Apply for Leave")

print("4. Create Pay Slip")

print("5. Show Projects")

print("6. Display All Employees")

print("7. Add Project")

print("8. Exit")

choice = input("Enter your choice: ")

if choice == "1":

add\_employee()

elif choice == "2":

remove\_employee()

elif choice == "3":

apply\_leave()

elif choice == "4":

create\_payslip()

elif choice == "5":

show\_projects()

elif choice == "6":

display\_all\_employees()

elif choice == "7":

add\_project()

elif choice == "8":

print("Exiting... Goodbye!")

break

else:

print("Invalid choice! Try again.\n")

menu()

## Conclusion

This project demonstrates how Python can be used to design a simple HR Management System. The functionalities implemented in the system provide the basic structure of real-world HR software. Working on this project improved our understanding of Python functions, data structures, and program design.