**EMPLOYEE MANAGEMENT SYSTEM**

(By Ankit Singh)

**INTRODUCTION**

1. Console-based Application: The project is a command-line-based Employee Management System that allows admins to manage employee data through a simple interface built with Python.

2. Admin Authentication: It features an admin registration and login system to secure access. Admin credentials are stored in a MySQL database, and the login system ensures that only registered users can access employee data.

3. CRUD Operations: The system enables admins to perform Create, Read, Update, and Delete (CRUD) operations on employee records. These include adding new employees, viewing employee details, updating specific information (like salary, Post, email, Name, Address), and deleting employee records.

4. MySQL Database Integration: All employee data and admin credentials are stored in a MySQL database, ensuring data persistence. The database contains tables for Employee(Id, Name, Email\_Id, Phone\_no , Address, Post, Salary) and admin information (Emp Id, Employee Email).

5. Structured Code Design: The project is designed with a modular approach, separating functionalities like authentication, employee management, and database connection, making the code organized and easy to maintain.

**COMPLETE CODE :-**

Employee\_management.py :-

from os import system

import re

from auth import authenticate\_user

import operation

def menu():

system("clear")

print("{:>60}".format("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"))

print("{:>60}".format("-->> Employee Management System <<--"))

print("{:>60}".format("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"))

print("1. Add Employee")

print("2. Display Employee Record")

print("3. Update Employee Record")

print("4. Promote Employee Record")

print("5. Remove Employee Record")

print("6. Search Employee Record")

print("7. Exit\n")

print("{:>60}".format("-->> Choice Options: [1/2/3/4/5/6/7] <<--"))

ch = int(input("Enter your Choice: "))

if ch == 1:

system("clear")

operation.Add\_Employ()

elif ch == 2:

system("clear")

operation.Display\_Employ()

elif ch == 3:

system("clear")

operation.Update\_Employ()

elif ch == 4:

system("clear")

operation.Promote\_Employ()

elif ch == 5:

system("clear")

operation.Remove\_Employ()

elif ch == 6:

system("clear")

operation.Search\_Employ()

elif ch == 7:

system("clear")

print("{:>60}".format("Have A Nice Day :)"))

exit(0)

else:

print("Invalid Choice!")

press = input("Press Any key To Continue..")

menu()

def main():

print("Welcome to the Employee Management System")

# -------

print("1. Log in")

print("2. Sign up")

choice = int(input("Enter your Choice: "))

if choice == 1:

if authenticate\_user(input("Enter Employee Email ID: "), input("Enter Employee ID: ")):

menu()

elif choice == 2:

operation.Add\_Employ()

else:

print("Authentication Failed. Exiting...")

exit(1)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Auth.py :-

import re

import dbs

# Email regex pattern

regex = r'\b[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'

def authenticate\_user(email\_id, employee\_id):

# Check if the email\_id is valid

if not re.fullmatch(regex, email\_id):

print("Invalid Email ID")

return False

sql = 'SELECT \* FROM empdata WHERE Email\_Id = %s AND Id = %s'

data = (email\_id, employee\_id)

c = dbs.con.cursor(buffered=True)

c.execute(sql, data)

if c.rowcount == 1:

return True

else:

print("Invalid Employee ID or Email ID")

return False

dbs.py :-

import mysql.connector

# Making Connection

con = mysql.connector.connect(

host="localhost", user="root", password="\*\*\*\*", database="Ankit")

operation.py :-

import re

import employee\_management

import dbs

# for validating an Email

regex = r'\b[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'

# for validating an Phone Number

Pattern = re.compile("(0|91)?[7-9][0-9]{9}")

def Add\_Employ():

print("{:>60}".format("-->>Add Employee Record<<--"))

Id = input("Enter Employee Id: ")

if (check\_employee(Id) == True):

print("Employee ID Already Exists\nTry Again..")

press = input("Press Any Key To Continue..")

employee\_management.Add\_Employ()

Name = input("Enter Employee Name: ")

if (check\_employee\_name(Name) == True):

print("Employee Name Already Exists\nTry Again..")

press = input("Press Any Key To Continue..")

employee\_management.Add\_Employ()

Email\_Id = input("Enter Employee Email ID: ")

if(re.fullmatch(regex, Email\_Id)):

print("Valid Email")

else:

print("Invalid Email")

press = input("Press Any Key To Continue..")

employee\_management.Add\_Employ()

Phone\_no = input("Enter Employee Phone No.: ")

if(Pattern.match(Phone\_no)):

print("Valid Phone Number")

else:

print("Invalid Phone Number")

press = input("Press Any Key To Continue..")

Add\_Employ()

Address = input("Enter Employee Address: ")

Post = input("Enter Employee Post: ")

Salary = input("Enter Employee Salary: ")

data = (Id, Name, Email\_Id, Phone\_no, Address, Post, Salary)

sql = 'insert into empdata values(%s,%s,%s,%s,%s,%s,%s)'

c = dbs.con.cursor()

c.execute(sql, data)

dbs.con.commit()

print("Successfully Added Employee Record")

press = input("Press Any Key To Continue..")

employee\_management.menu()

def check\_employee\_name(employee\_name):

sql = 'select \* from empdata where Name=%s'

c = dbs.con.cursor(buffered=True)

data = (employee\_name,)

c.execute(sql, data)

r = c.rowcount

if r == 1:

return True

else:

return False

def check\_employee(employee\_id):

sql = 'select \* from empdata where Id=%s'

c = dbs.con.cursor(buffered=True)

data = (employee\_id,)

c.execute(sql, data)

r = c.rowcount

if r == 1:

return True

else:

return False

def Display\_Employ():

print("{:>60}".format("-->> Display Employee Record <<--"))

sql = 'select \* from empdata'

c = dbs.con.cursor()

c.execute(sql)

r = c.fetchall()

for i in r:

print("Employee Id: ", i[0])

print("Employee Name: ", i[1])

print("Employee Email Id: ", i[2])

print("Employee Phone No.: ", i[3])

print("Employee Address: ", i[4])

print("Employee Post: ", i[5])

print("Employee Salary: ", i[6])

print("\n")

press = input("Press Any key To Continue..")

employee\_management.menu()

def Update\_Employ():

print("{:>60}".format("-->> Update Employee Record <<--\n"))

Id = input("Enter Employee Id: ")

if(check\_employee(Id) == False):

print("Employee Record Not exists\nTry Again")

press = input("Press Any Key To Continue..")

employee\_management.menu()

else:

Email\_Id = input("Enter Employee Email ID: ")

if(re.fullmatch(regex, Email\_Id)):

print("Valid Email")

else:

print("Invalid Email")

press = input("Press Any Key To Continue..")

Update\_Employ()

Phone\_no = input("Enter Employee Phone No.: ")

if(Pattern.match(Phone\_no)):

print("Valid Phone Number")

else:

print("Invalid Phone Number")

press = input("Press Any Key To Continue..")

Update\_Employ()

Address = input("Enter Employee Address: ")

name = input("Enter Employee name: ")

sql = 'UPDATE empdata set Email\_Id = %s, Phone\_no = %s, Address = %s, Name = %s where Id = %s'

data = (Email\_Id, Phone\_no, Address, name, Id)

c = dbs.con.cursor()

c.execute(sql, data)

dbs.con.commit()

print("Updated Employee Record")

press = input("Press Any Key To Continue..")

employee\_management.menu()

def Promote\_Employ():

print("{:>60}".format("-->> Promote Employee Record <<--\n"))

Id = input("Enter Employee Id: ")

if(check\_employee(Id) == False):

print("Employee Record Not exists\nTry Again")

press = input("Press Any Key To Continue..")

employee\_management.menu()

else:

Amount = int(input("Enter Increase Salary: "))

sql = 'select Salary from empdata where Id=%s'

data = (Id,)

c = dbs.con.cursor()

c.execute(sql, data)

r = c.fetchone()

t = r[0]+Amount

sql = 'update empdata set Salary = %s where Id = %s'

d = (t, Id)

c.execute(sql, d)

dbs.con.commit()

print("Employee Promoted")

press = input("Press Any key To Continue..")

employee\_management.menu()

def Remove\_Employ():

print("{:>60}".format("-->> Remove Employee Record <<--\n"))

Id = input("Enter Employee Id: ")

if(check\_employee(Id) == False):

print("Employee Record Not exists\nTry Again")

press = input("Press Any Key To Continue..")

employee\_management.menu()

else:

sql = 'delete from empdata where Id = %s'

data = (Id,)

c = dbs.con.cursor()

c.execute(sql, data)

dbs.con.commit()

print("Employee Removed")

press = input("Press Any key To Continue..")

employee\_management.menu()

def Search\_Employ():

print("{:>60}".format("-->> Search Employee Record <<--\n"))

Id = input("Enter Employee Id: ")

if(check\_employee(Id) == False):

print("Employee Record Not exists\nTry Again")

press = input("Press Any Key To Continue..")

employee\_management.menu()

else:

sql = 'select \* from empdata where Id = %s'

data = (Id,)

c = dbs.con.cursor()

c.execute(sql, data)

r = c.fetchall()

for i in r:

print("Employee Id: ", i[0])

print("Employee Name: ", i[1])

print("Employee Email Id: ", i[2])

print("Employee Phone No.: ", i[3])

print("Employee Address: ", i[4])

print("Employee Post: ", i[5])

print("Employee Salary: ", i[6])

print("\n")

press = input("Press Any key To Continue..")

employee\_management.menu()

Outputs :-

Welcome to the Employee Management System

1. Log in

2. Sign up

Enter your Choice: 1

Enter Employee Email ID: ankit@gmail.com

Enter Employee ID: 1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**-->> Employee Management System <<--**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Add Employee

2. Display Employee Record

3. Update Employee Record

4. Promote Employee Record

5. Remove Employee Record

6. Search Employee Record

7. Exit

-->> Choice Options: [1/2/3/4/5/6/7] <<--

Enter your Choice:

**-->> Display Employee Record <<--**

Employee Id: 1

Employee Name: Ankit Singh

Employee Email Id: ankit@gmail.com

Employee Phone No.: 9876543210

Employee Address: Kharadi, Pune

Employee Post: Software Developer

Employee Salary: 60000.0

Press Any key To Continue..

**-->> Update Employee Record <<--**

Enter Employee Id: 1

Enter Employee Email ID: ankits@gmail.com

Valid Email

Enter Employee Phone No.: 9876789654

Valid Phone Number

Enter Employee Address: Kashmiri Gat, Delhi

Enter Employee name: Ankit Pratap

Updated Employee Record

Press Any Key To Continue..

**-->> Promote Employee Record <<--**

Enter Employee Id: 1

Enter Increase Salary: 4000

Employee Promoted

Press Any key To Continue..

**-->> Remove Employee Record <<--**

Enter Employee Id: 1

Employee Record Not exists

Try Again

Press Any Key To Continue..

**-->> Search Employee Record <<--**

Enter Employee Id: 1

Employee Id: 1

Employee Name: Ankit Pratap

Employee Email Id: ankits@gmail.com

Employee Phone No.: 9876789654

Employee Address: Kashmiri Gat, Delhi

Employee Post: Software Developer

Employee Salary: 64000.0

Press Any key To Continue..

**Have A Nice Day :)**