**Acknowledgement**

*I am over helmed in all humbleness and gratefulness to acknowledge my depth to all those who helped me to put these ideas, well above the level of simplicity and into something concrete.*

*I would like to express my special thanks of gratitude to my Computer Science Teacher Mr. Shahbaz Ali for his guidance and constant motivation and help. I would also like to thank my parents and friends for helping me completing this project within the limited time. I am making this project not only for marks but also to increase my knowledge.*

*Thanks again to all who have helped me.*

***Class: XII-PCM***

**Introduction**

Employee Records Management System is very beneficial to perform Employee Management related tasks easily and efficiently. It saves a lot of time by performing tasks in the system rather than maintaining registers.

It assists to keep track of employee’s information such as salary details, medical related information, attendance and much more.

The recommended software and hardware requirements are easily available. Maintaining registers for this can be a complex task. Employee Record Management System is an error free, secure, reliable management system. Thus, it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means no one need not to be distracted by information that is not relevant, while being able to reach the information.

**Hardware and Software**

**Used:**

Hardware:

* PROCESSOR – Intel i5-4210U CPU @ 1.70GHz 2.40GHz
* RAM – 4GB
* OS – WINDOWS 10
* STORAGE – 500GB SSHD
* PRINTER – To Print the Required Documents

Software:

* Python 3.10.8
* MySQL 8.0.3
* Microsoft Word 365 (Output Presentation)

Recommended:

Hardware:

* PROCESSOR – PENTIUM DUAL CORE
* RAM – 2GB
* OS – WINDOWS 7(SERVICE PACK 1) OR WINDOWS 10
* STORAGE – 80GB

Software:

* Python 3.2 or Later
* MySQL 5.6 or Later
* Microsoft Word 2007 or Later
* Pen drive/Compact Disc (CD)

**Modules**

* **OS Module**

The python OS module provides function used for interacting with the operating system and get related information about it.

* **RE Module**

Short name – regular expression, is used for various string patterns-that is the regex-a string to search against and return a result usually a Match object.

* **MYSQL-Connector**

MySQL Connector enables python program to access MySQL databases.

**Coding**

**#Importing Required Modules for the program**

from os import system

import re

**#Importing mysql connector**

import mysql.connector

**#Creating database**

def create\_database():

con = mysql.connector.connect(host="localhost", user="root", password="224783")

c = con.cursor()

c.execute("CREATE DATABASE IF NOT EXISTS EMPLOYEE;")

create\_database()

**#Creating table**

def create\_table():

con = mysql.connector.connect(host="localhost", user="root", password="224783", database="Employee")

c = con.cursor()

c.execute("SHOW TABLES;")

data = c.fetchall()

for i in data:

if(i == ('empdata',)):

break

else:

c.execute("CREATE TABLE empdata(Id INT(11) PRIMARY KEY, Name VARCHAR(1000), Gender VARCHAR(7), Email\_Id TEXT(1000), Phone\_no VARCHAR(15), Address TEXT(1000), Post TEXT(1000), Salary BIGINT(20))")

create\_table()

**#Making Connection**

con = mysql.connector.connect(host="localhost", user="root", password="224783", database="employee")

**#Make a regular expression for validating an Email**

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

**#Function to Add\_Employ**

def Add\_Employ():

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

Id = input("Enter Employee Id: ")

**#Checking If Employee Id is Exit or Not**

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

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

press = input("Press Any key to continue...")

Add\_Employ()

Name = input("Enter Employee Name: ")

Gender = input("Enter Employee Gender: ")

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

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

print("Valid Email, Please proceed")

else:

print("Invalid Email")

press = input("Press Any key to continue...")

menu()

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

Address = input("Enter Employee Address: ")

Post = input("Enter Employee Post: ")

Salary = input("Enter Employee Salary: ")

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

**#Inserting Employee Details in empdata**

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

c = con.cursor()

c.execute(sql, data)

con.commit()

print("Successfully Added Employee Record")

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

menu()

**#Function to check if Employee with given Id exist or not**

def check\_employee(employee\_id):

#query to select all rows from employee(empdata) table

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

#making cursor buffered to make rowcount method work properly

c = con.cursor(buffered=True)

data = (employee\_id,)

c.execute(sql, data)

**#rowcount method to find number of rows with given values**

r = c.rowcount

if r == 1:

return True

else:

return False

**#Function to Display\_Employ**

def Display\_Employ():

print("{:>60}".format("-->>Displaying All Employee Records<<--"))

**#query to select all rows from Employee (empdata) Table**

sql = 'select \* from empdata'

c = con.cursor()

c.execute(sql)

**#Fetching all details of all the Empoyees**

r = c.fetchall()

for i in r:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

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

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

print("Employee Gender: ", i[2])

print("Employee Email Id: ", i[3])

print("Employee Phone No: ", i[4])

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

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

print("Employee Salary: ", i[7]) print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\n")

press = input("Press Any key to continue...")

menu()

**#Function to Update\_Employ**

def Update\_Employ():

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

Id = input("Enter Employee Id: ")

**#Checking If Employee Id is Exit or Not**

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

print("Employee ID Does Not Exists\nTry Again...")

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

menu()

else:

print("What do you want to Update? ")

print("1. Name")

print("2. Gender")

print("3. Email ID")

print("4. Phone No")

print("5. Address")

print("6. Post")

print("7. Salary")

ch = int(input("Enter your Update Preference Number from Above (1,2,3,4,5,6,7): "))

if ch == 1:

Name = input("Enter Employee Name: ")

sql = 'UPDATE empdata set Name = %s where Id = %s'

data = (Name, Id)

c = con.cursor()

c.execute(sql, data)

con.commit()

print("Updated Employee Name")

print("1. Want to Update more Records\n2. Go back to Menu")

cho = int(input("Press 1 or 2: "))

if cho == 1:

Update\_Employ()

else:

menu()

elif ch == 2:

Gender = input("Enter Employee Gender: ")

sql = 'UPDATE empdata set Gender = %s where Id = %s'

data = (Gender, Id)

c = con.cursor()

c.execute(sql, data)

con.commit()

print("Updated Employee Gender")

print("1. Want to Update more Records\n2. Go back to Menu")

cho = int(input("Press 1 or 2: "))

if cho == 1:

Update\_Employ()

else:

menu()

elif ch == 3:

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

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

print("Valid Email, Please proceed")

else:

print("Invalid Email")

press = input("Press Any key to continue...")

menu()

sql = 'UPDATE empdata set Email\_Id = %s where Id = %s'

data = (Email\_Id, Id)

c = con.cursor()

c.execute(sql, data)

con.commit()

print("Updated Employee Email\_Id")

print("1. Want to Update more Records\n2. Go back to Menu")

cho = int(input("Press 1 or 2: "))

if cho == 1:

Update\_Employ()

else:

menu()

elif ch == 4:

Phone\_no = input("Enter Employee Phone\_no: ")

sql = 'UPDATE empdata set Phone\_no = %s where Id = %s'

data = (Phone\_no, Id)

c = con.cursor()

c.execute(sql, data)

con.commit()

print("Updated Employee Phone\_no")

print("1. Want to Update more Records\n2. Go back to Menu")

cho = int(input("Press 1 or 2: "))

if cho == 1:

Update\_Employ()

else:

menu()

elif ch == 5:

Address = input("Enter Employee Address: ")

sql = 'UPDATE empdata set Address = %s where Id = %s'

data = (Address, Id)

c = con.cursor()

c.execute(sql, data)

con.commit()

print("Updated Employee Address")

print("1. Want to Update more Records\n2. Go back to Menu")

cho = int(input("Press 1 or 2: "))

if cho == 1:

Update\_Employ()

else:

menu()

elif ch == 6:

Post = input("Enter Employee Post: ")

sql = 'UPDATE empdata set Post = %s where Id = %s'

data = (Post, Id)

c = con.cursor()

c.execute(sql, data)

con.commit()

print("Updated Employee Post")

print("1. Want to Update more Records\n2. Go back to Menu")

cho = int(input("Press 1 or 2: "))

if cho == 1:

Update\_Employ()

else:

menu()

elif ch == 7:

Salary = input("Enter Employee Salary: ")

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

data = (Salary, Id)

c = con.cursor()

c.execute(sql, data)

con.commit()

print("Updated Employee Salary")

print("1. Want to Update more Records\n2. Go back to Menu")

cho = int(input("Press 1 or 2: "))

if cho == 1:

Update\_Employ()

else:

menu()

else:

print("Invalid Choice\nTry Again")

menu()

**#Function to Promote Employ**

def Promote\_Employ():

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

Id = input("Enter Employee Id: ")

#Checking If Employee Id is Exit or Not

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

print("Employee ID Does Not Exists\nTry Again...")

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

menu()

else:

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

#query to fetch salary of Employee with given data

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

data = (Id,)

c = con.cursor()

#executing the sql query

c.execute(sql, data)

**#fetching salary of employee with given Id**

r = c.fetchone()

t = r[0]+amount

**#query to update salary of employee with given id**

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

d = (t, Id)

**#executing the sql query**

c.execute(sql, d)

**#commit() method to make changes in the table**

con.commit()

print("Employee Promoted")

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

menu()

**#Function to Remove\_Employ**

def Remove\_Employ():

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

Id = input("Enter Employee Id: ")

**#Checking If Employee Id is Exit or Not**

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

print("Employee ID Does Not Exists\nTry Again...")

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

menu()

else:

**#query to delete Employee from empdata table**

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

data = (Id,)

c = con.cursor()

**#executing the sql query**

c.execute(sql, data)

**#commit method to make changes in the empdata table**

con.commit()

print("Employee Record Removed")

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

menu()

**#Function to Search\_Employ**

def Search\_Employ():

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

Id = input("Enter Employee Id: ")

**#Checking If Employee Id is Exit or Not**

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

print("Employee ID Does Not Exists\nTry Again...")

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

menu()

else:

#query to search Employee from empdata table

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

data = (Id,)

c = con.cursor()

**#executing the sql query**

c.execute(sql, data)

**#fetching all details of all the employees**

r = c.fetchall()

for i in r:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

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

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

print("Employee Gender: ", i[2])

print("Employee Email Id: ", i[3])

print("Employee Phone No: ", i[4])

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

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

print("Employee Salary: ", i[7])

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\n")

press = input("Press Any key to continue...")

menu()

**#Menu function to display menu**

def menu():

system("cls")

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

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

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

print("1. Add Employee")

print("2. Display All Employee Record")

print("3. Update Employee Record")

print("4. Promote Employee")

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("cls")

Add\_Employ()

elif ch == 2:

system("cls")

Display\_Employ()

elif ch == 3:

system("cls")

Update\_Employ()

elif ch == 4:

system("cls")

Promote\_Employ()

elif ch == 5:

system("cls")

Remove\_Employ()

elif ch == 6:

system("cls")

Search\_Employ()

elif ch == 7:

system("cls")

print("{:>60}".format("Exiting..."))

else:

print("Invalid Choice!")

press = input("Press Any key to continue...")

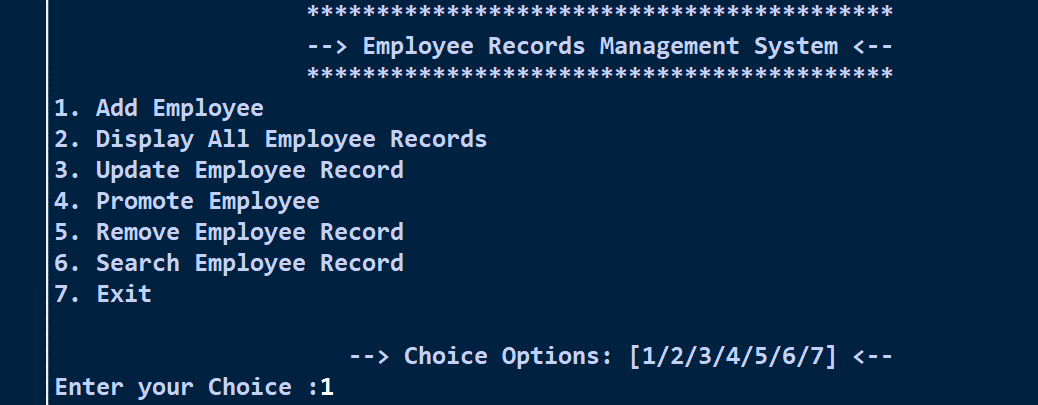
menu()

**#Calling menu funtion**

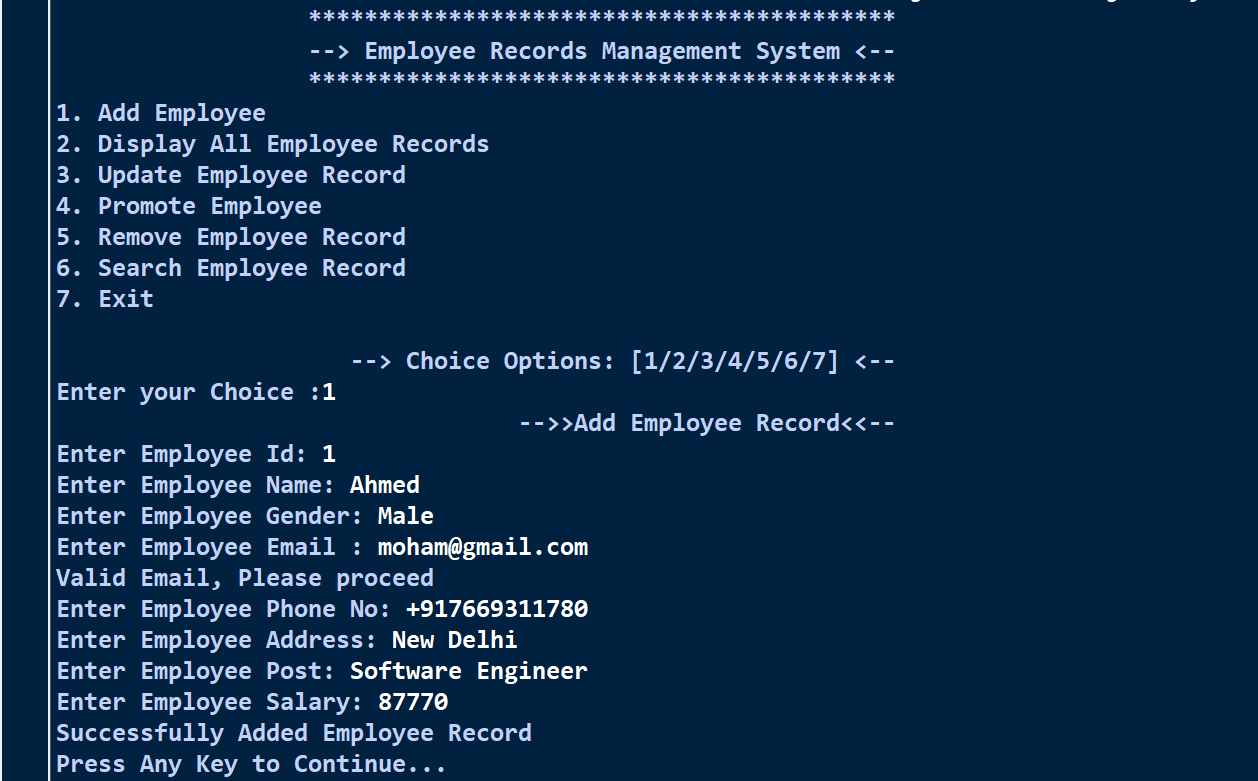
menu()

**Output**

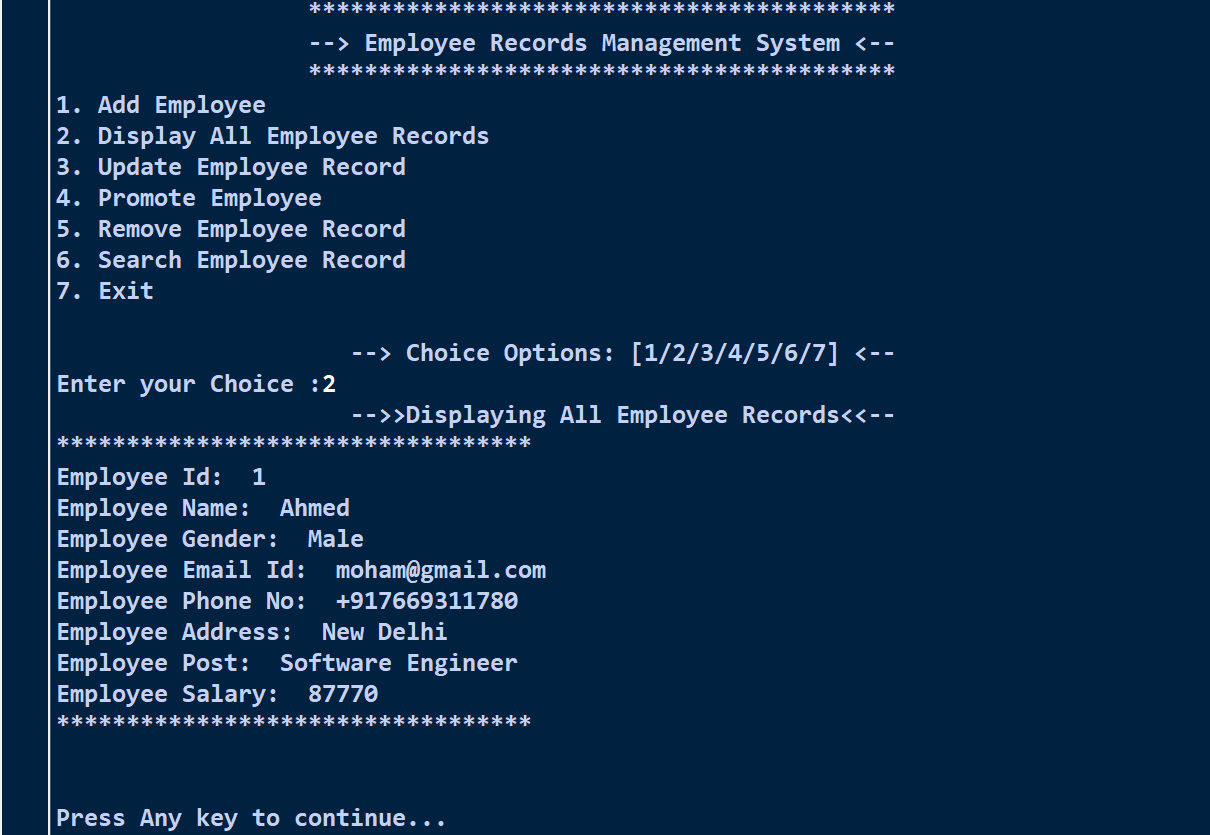
1. **Main Menu**

****

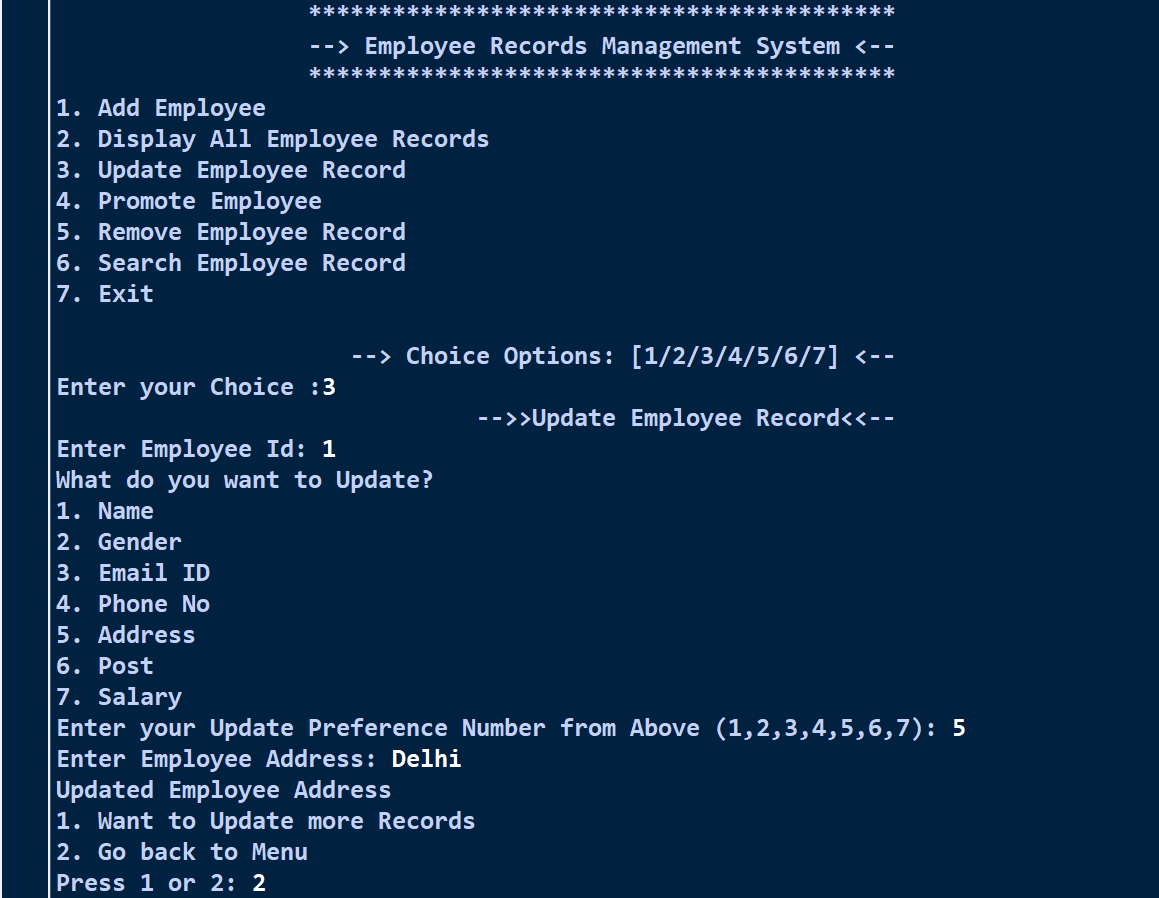
1. **Adding Employee**

****

1. **Displaying All Records**

****

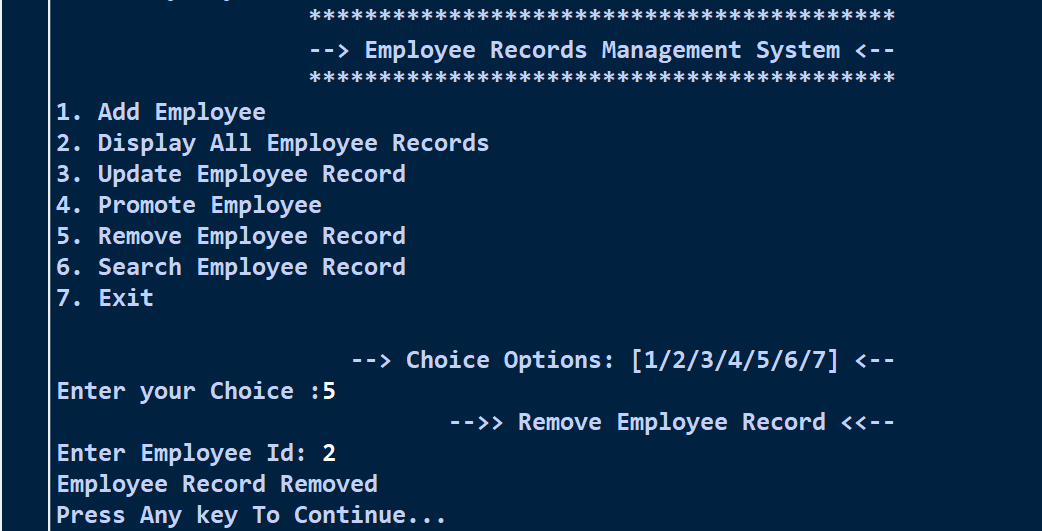
1. **Updating Employee Record**

****

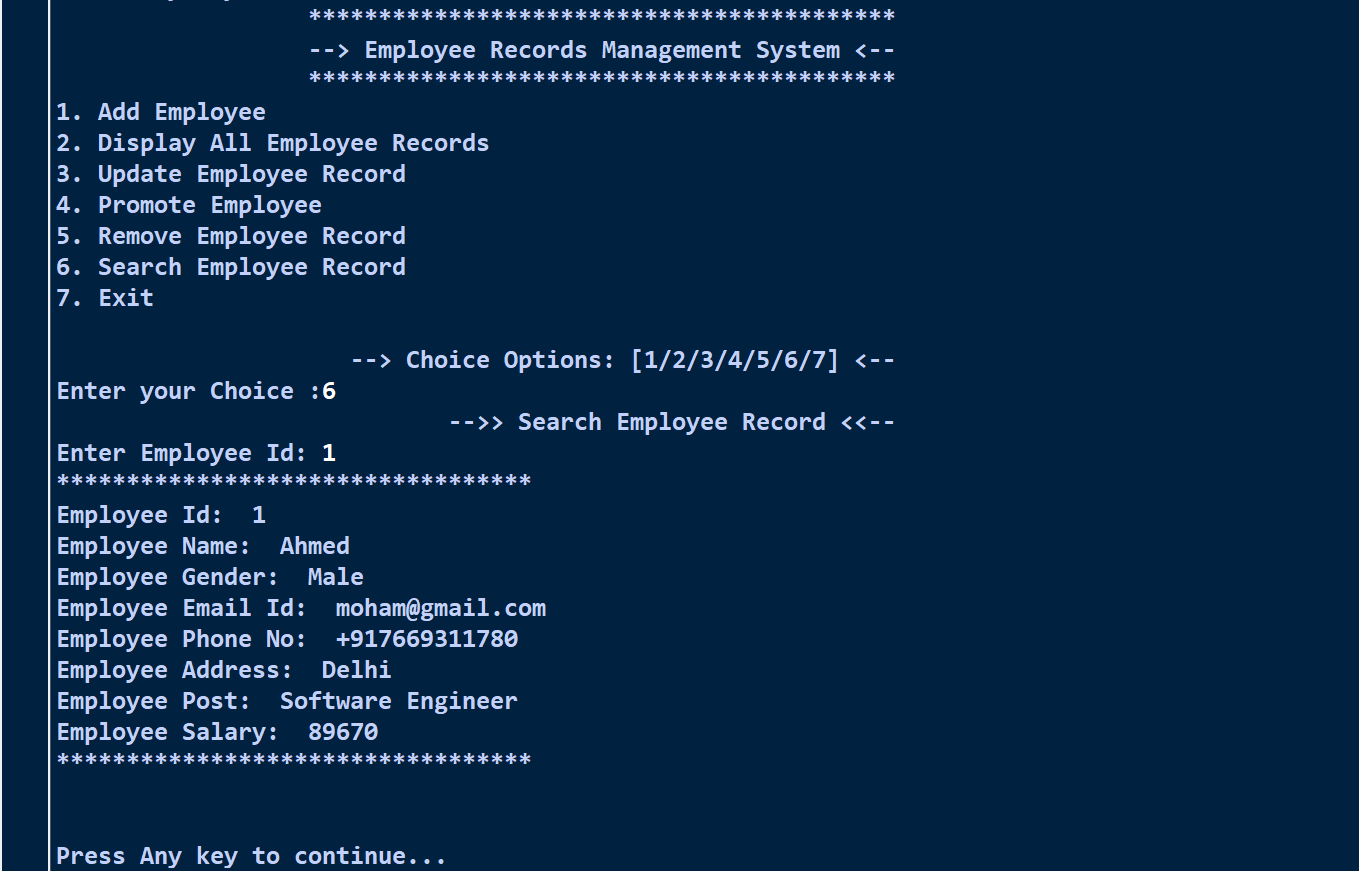
1. **Promoting Employee**

****

1. **Removing Employee**

****

1. **Search Employee Record**

****

**References**

* Computer Science with Python Class XII by Sumit Arora.
* Stack Overflow -<https://stackoverflow.com>
* W3Schools –

<https://www.w3schools.com>