

INDEX

S. NO.	CONTENTS	PAGE
1.	CERTIFICATE	3
2.	ACKNOWLEDGMENT	4
3.	SYNOPSIS	5
4.	BUILT-IN MODULES/FUNCTIONS USED	10
5.	USER DEFINED MODULES & FUNCTIONS USED	10
6.	DATABASE & TABLES USED	11
7.	SOFTWARE & HARDWARE SPECIFIC INFORMATION	12
8.	SOURCE CODE	13
9.	OUTPUT SCREENS/REPORTS	66
10.	LIMITATIONS	81
11.	BIBLIOGRAPHY	82



AHLCON PUBLIC SCHOOL

Dept of Computer Science



CERTIFICATE

This is to certify that Akshat Singh of class XII-D has prepared a project entitled 'EMPLOYEE MANAGEMENT SYSTEM'. This project is the result of his/her effort and endeavors. This project is found worthy of acceptance as a final Project Report for the subject Computer Science of Class XII. Aryan has prepared the project under my Guidance.

Mr Jagdish Prasad
Lecturer in Computer Science
Ahlcon Public School
Mayur Vihar Ph 1 Delhi 91

ACKNOWLEDGEMENT

In the accomplishment of this project successfully, many people have best owned upon me their blessings and the heart pledged support, this time I am utilizing to thank all the people who have been concerned with project. Primarily I would thank god for being able to complete this project with success. Then I would like to thank my principal Dr. Deepak Bisht and Computer Science teacher Mr. Jagdish Prasad, whose valuable guidance has been the ones that helped me patch this project and make it full proof success his suggestions and her instructions has served as the major contributor towards the completion of the project. Then I would like to thank my parents and friends who have helped me with their valuable suggestions and guidance has been helpful in various phases of the completion of the project.

Akshat Singh

12-D

SYNOPSIS

OVERVIEW

Python is an interpreted, high-level and general-purpose programming language used worldwide. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

MySQL is an open-source relational database management system (RDBMS). A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmer use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

A **payroll management system** is a tool - predominantly a **software** program - that enables your business to handle all your **employee's** financial records in a hassle-free, automated fashion. This includes **employee's** salaries, bonuses, deductions, net pay, and generation of pay-slips for a specific period.

In this project, we have used “**Python-SQL Interface**” to design a **EMPLOYEE DATABASE MANGEMENT**.

CHRONOLOGICAL ORDER OF WORKING

This system finds its use in Professional Organizations. The administrator can pre-feed the information of the employees and their pay checks and the relevant commands with its regard for easier access. The program proves to be of much use in managing details of employees and displaying their pay checks. Name of the entire database is “emp_database”.

There are two tables which have been created under this database which are as follows:

- The administrative table (can be accessed by only the admin)
- The personal table (can be accessed by the employees of the organization)

The Two tables have interlinked access which is in the chronological procedure. However, the table for Administrator has a special access procedure which preserves the security of the system. So, the User knowing the Security Access Password can only view and edit that table. Access to each table leads to further access to chronological details for issuing a particular command.

The beginning procedure by choosing whether you are the admin or an employee and for accessing the administrative table you need the password - ‘admin’. However, the employees don’t need a password to access the personal table.

The admin has various controls over the database and edit, delete or amend the information present in the database.

The employee can edit his/her personal information and can view their pay check and the remarks assigned to them.

Note: False identification of the User may generate an error. For smooth access to the system it is appreciated to wait for the access time from the User input till the output generated. After completing the respective task, the user gets an option to continue or close the program as per his needs. The User by typing “no” when prompted can exit the program. Selecting YES, will lead the User to further keep working on the database. To re-enter the User has to go through the initial procedures again.

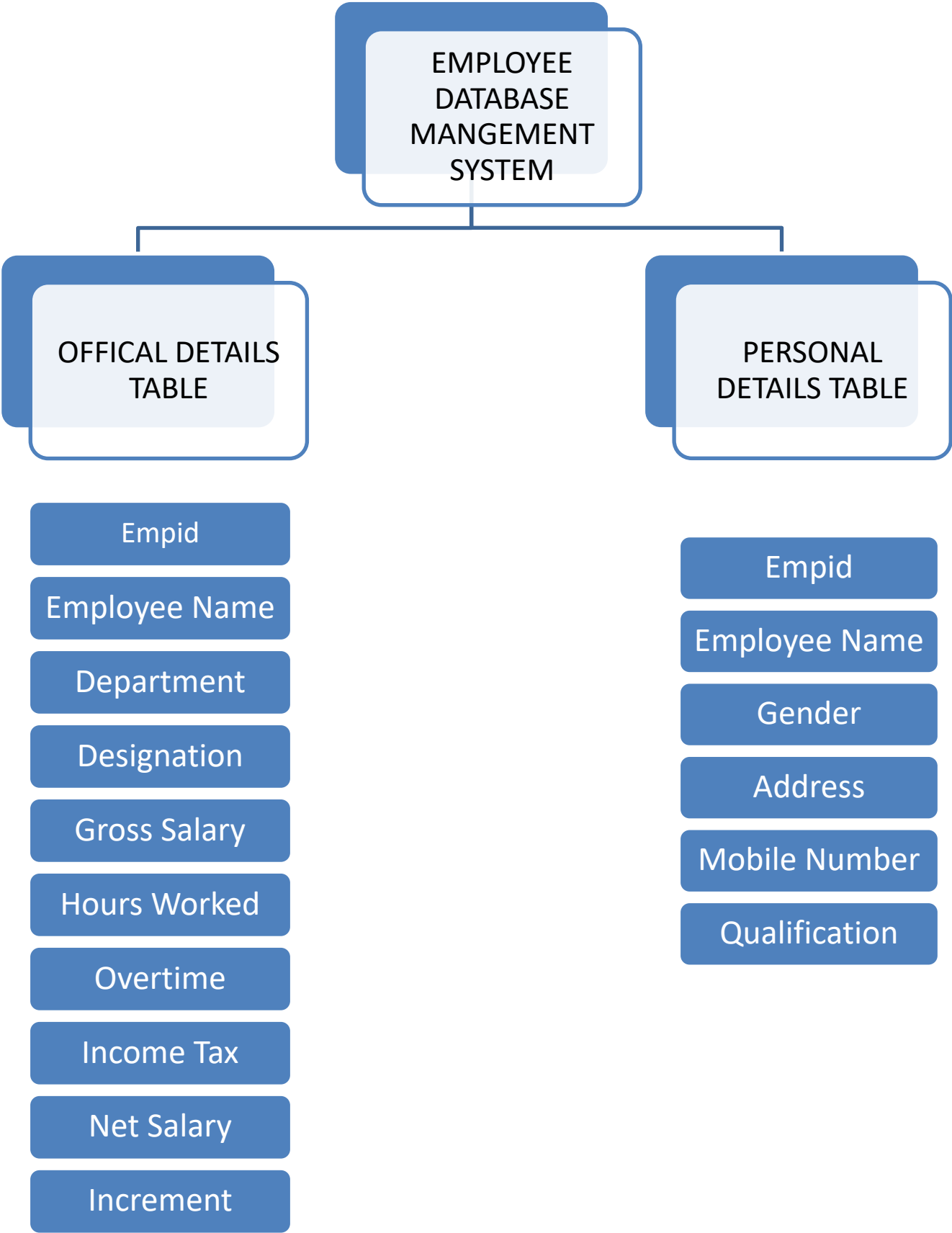
ROUND ABOUT OF THE PROGRAM

It is a general- access program. According to User type the procedure will begin. In case the User is the Administrator, then the Security procedural commands will ask for accessing the ‘Administrator Details’ with prior ID-password authentication. From there on that User can choose to emend the Official Details or Edit the various personal details of employees submitted into the database.

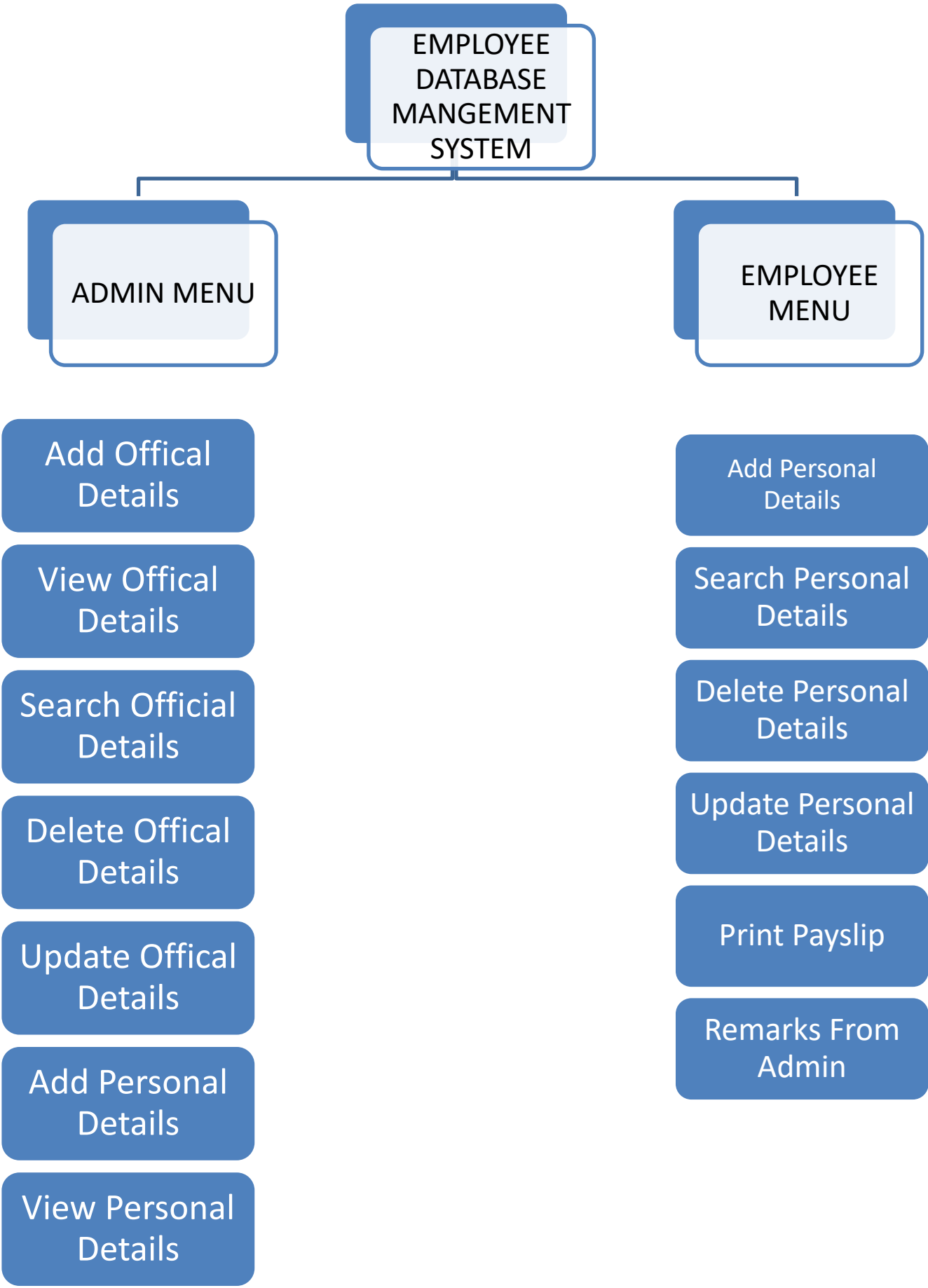
In case the User is an Employee- the system will ask whether to change or view his/her details and thereby will provide the output for choosing the preferred action from the table.

The employee can ultimately view their pay check and get to know their remarks as per entered in the database.

SEQUENTIAL TABLE



SEQUENTIAL MENU



USER DEFINED

MOUDLES/FUNCTIONS USED

- I. check_connection
- II. create_database
- III. add_new_record
- IV. display_all_record
- V. search_record
- VI. delete_recod
- VII. update_record
- VIII. add_new_record_personal_admin
- IX. display_record_personal
- X. add_new_record_persona
- XI. search_record_personal
- XII. delete_record_personal
- XIII. update_record_personal
- XIV. payroll_slip_data_from_personal_table
- XV. payroll_slip
- XVI. emp_recomendation

BUILT IN MODULES/FUNCTIONS USED

- I. random Module
- II. random.choice Function
- III. mysql.connector Module

DATABASE & TABLES USED:

The name of the database used in this program is “emp_database”.

The Program contains 2 tables: emp_payroll and emp_personal.

- Emp_Payroll:

empid	employeename	department	designation	gross_salary	hours_worked	overtime_pay	income_tax	net_salary	increment
101	ayush	finance	manager	25000.0	45	6250.0	1250.0	30000.0	0.0
102	bani	consultancy	assistant	15000.0	50	7500.0	750.0	21750.0	0.0
103	jacob	IT	engineer	50000.0	40	0.0	2500.0	47500.0	0.0
104	ranjan	finance	accountant	30000.0	45	7500.0	1500.0	36000.0	0.0
105	aman	finance	manager	24000.0	48	9600.0	1200.0	32400.0	0.0
106	jaya	IT	assistance	15000.0	38	-1500.0	750.0	12750.0	0.0

- Emp_Personal:

empid	employeename	gender	address	mobile_number	qualification
101	ayush	male	FARIDABAD	7755687421	BBA+MBA
102	bani	female	sector-3,Noida	9852123102	MBA
103	jacob	male	Phase 1,Delhi	8856542130	Btech
104	ranjan	male	meerut	9865321245	CA
105	aman	male	noida	8552265324	MBA
106	jaya	female	delhi	7025642315	Btech

HARDWARE SPECIFIC INFO



CPU- INTEL CORE I5-10600K



GPU-NVIDIA GTX 1660TI



RAM-8 GB GDDR4



SSD-256 GB

SOFTWARE SPECIFIC INFO



OS: WINDOWS 10 (64 BIT)



PYTHON IDLE (3.7.2)



MySQL (8.0)

SOURCE CODE

```
import random
```

```
import mysql.connector
```

```
mycon=None
```

```
mycursor=None
```

```
flag=0
```

```
sp=""
```

```
def check_connection():
```

```
    mycon = mysql.connector.connect(
```

```
        host="localhost",
```

```
        user="root",
```

```
        passwd="somerville"
```

```
    )
```

```
    if mycon.is_connected():
```

```
        print("Successfully connected to MySQL")
```

```
        flag=1
```

```
    return flag
```

```
def create_database():
```

```
    mycon = mysql.connector.connect(  
        host="localhost",  
        user="root",  
        passwd="somerville"  
    )
```

```
    mycursor = mycon.cursor()
```

```
    mycursor.execute("CREATE DATABASE IF NOT EXISTS emp_database")
```

```
    print("Database created or used")
```

```
def create_table():
```

```
    mycon = mysql.connector.connect(  
        host="localhost",  
        user="root",  
        passwd="somerville",  
        database="emp_database"  
    )
```

```
    mycursor = mycon.cursor()
```

```
mycursor.execute("""CREATE TABLE IF NOT EXISTS emp_payroll(empid int primary key  
auto_increment ,employeename varchar(20),department varchar(15),designation  
varchar(20),gross_salary float(10),hours_worked int(5),overtime_pay float(10),income_tax float(10)  
    ,net_salary float(10),increment float(10))""")
```

```
def add_new_record():
```

```
    mycon = mysql.connector.connect(
```

```
        host="localhost",
```

```
        user="root",
```

```
        passwd="somerville",
```

```
        database="emp_database"
```

```
    )
```

```
    mycursor = mycon.cursor()
```

```
    empid=int(input("ENTER EMPLOYEE ID:\t"))
```

```
    employeename=input("ENTER EMPLOYEE NAME:\t")
```

```
    department=input("ENTER DEPARTMENT OF EMPLOYEE:\t")
```

```
    designation=input("ENTER DESIGNATION OF EMPLOYEE\t")
```

```
    gross_salary=float(input("ENTER GROSS SALARY OF THE EMPLOYEE\t"))
```

```
    hours_worked=int(input("ENTER TOTAL HOURS WORKED\t"))
```

```
dif_hour=int(hours_worked-40)
```

```
op=((0.05*gross_salary)*dif_hour)
```

```
income_tax=float(0.05*gross_salary)
```

```
net_salary=float(gross_salary+op-income_tax)
```

```
increment=float('0')
```

```
sql = '''INSERT INTO
```

```
emp_payroll(empid,employeename,department,designation,gross_salary,hours_worked,overtime_
pay,income_tax,net_salary,increment)
```

```
VALUES({},'{}','{}','{}',{},{},{},{},{},{})'''.format(empid,employeename,department,designation,gross_sal
ary,hours_worked,op,income_tax,net_salary,increment)
```

```
mycursor.execute(sql)
```

```
mycon.commit()
```

```
print(mycursor.rowcount, "record inserted.")
```

```
def space(V):
```

```
    global sp
```

```
    sp=""
```

```
    l=15-len(str(V))
```

```
    for i in range(l):
```

```
sp=sp+" "
```

```
return sp
```

```
def display_all_record():
```

```
    mycon = mysql.connector.connect(
```

```
        host="localhost",
```

```
        user="root",
```

```
        passwd="somerville",
```

```
        database="emp_database"
```

```
    )
```

```
    mycursor = mycon.cursor()
```

```
    mycursor.execute("SELECT * FROM emp_payroll")
```

```
    myresult = mycursor.fetchall()
```

```
    print(" |-----|")
    print("-----|")
```

```
    print(" | empid",space("empid"),"employeeename",
space("employeeename"),"department",space("department"),"designation",space("designation"),"gr
oss_salary",space("gross_salary"),"hours_worked",space("hours_worked"),"overtime_pay",end=" ")
```

```
print(space("overtime_pay"),"income_tax",space("income_tax"),"net_salary",space('net_salary'),'inc
rement',space('increment')," |")
```



```
print("|-----|")
-----|")
```

```
for x in myresult:
```

```
print(' ',x[0],space(str(x[0])),x[1],space(x[1]),x[2],space(x[2]),x[3],space(x[3]),x[4],space(str(x[4])),x[5]
,space(str(x[5])),x[6],space(str(x[6])),x[7],space(str(x[7])),x[8],space(str(x[8])),x[9],space(str(x[9])),' ')
```

```
print("|-----|")
-----|")
```

```
def search_record():
```

```
mycon = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="somerville",
    database="emp_database"
)
```

```
mycursor = mycon.cursor()
```

```
print("""
```

```
*****
```

```
*      TO SERACH USING EMPLOYEE NAME      1      *
```

```
*      TO SEARCH USING EMPLOYEE ID      2      *
```

*	TO SEARCH USING DESIGNATION	3	*
*	TO SEARCH USING DEPARTMENT	4	*
*	TO SEARCH USING GROSS SALARY	5	*

*****")

print("")

detail=input('ENTER YOUR CHOICE:\t')

if (detail=='1'):

print("")

NAME=input("Enter Employee name to search:\t")

print("")

sql_select_query = "select * from emp_payroll where employeename = %s"

mycursor.execute(sql_select_query, (NAME,))

myresult = mycursor.fetchall()

if (mycursor.rowcount>0):

print(" |-----
-----|")

```

        print(" | empid",space("empid"),"employee name",
space("employee name"),"department",space("department"),"designation",space("designation"),"gr
oss_salary",space("gross_salary"),"hours_worked",space("hours_worked"),"overtime_pay",end=" ")

print(space("overtime_pay"),"income_tax",space("income_tax"),"net_salary",space('net_salary'),'inc
rement',space('increment')," |")

        print(" |-----
-----|")

        for x in myresult:

print(' | ',x[0],space(str(x[0])),x[1],space(x[1]),x[2],space(x[2]),x[3],space(x[3]),x[4],space(str(x[4])),x[5]
,space(str(x[5])),x[6],space(str(x[6])),x[7],space(str(x[7])),x[8],space(str(x[8])),x[9],space(str(x[9])),' | ')

        print(" |-----
-----|")

        else:

        print("No employee found")

elif (detail=='2'):

        print("")

        ID=int(input("Enter Employee id to search:\t"))

        print("")

        sql_select_query = "select * from emp_payroll where empid = %s"

        mycursor.execute(sql_select_query, (ID, ))

```

```
myresult = mycursor.fetchall()
```

```
if (mycursor.rowcount>0):
```

```
    print(" |-----|")
```

```
        print(" | empid",space("empid"),"employeenam",  
space("employeenam"),"department",space("department"),"designation",space("designation"),"gr  
oss_salary",space("gross_salary"),"hours_worked",space("hours_worked"),"overtime_pay",end=" ")
```

```
print(space("overtime_pay"),"income_tax",space("income_tax"),"net_salary",space('net_salary'),'inc  
rement',space('increment')," |")
```

```
    print(" |-----|")
```

```
for x in myresult:
```

```
print(' |',x[0],space(str(x[0])),x[1],space(x[1]),x[2],space(x[2]),x[3],space(x[3]),x[4],space(str(x[4])),x[5]  
,space(str(x[5])),x[6],space(str(x[6])),x[7],space(str(x[7])),x[8],space(str(x[8])),x[9],space(str(x[9])),' |')
```

```
    print(" |-----|")
```

```
else:
```

```
    print("No employee found")
```

```
elif (detail=='3'):
```

```
    print("")
```

```
DESIG=input("Enter Designation to search:\t")
```

```
print("")
```

```
sql_select_query = "select * from emp_payroll where designation = %s"
```

```
mycursor.execute(sql_select_query, (DESIG, ))
```

```
myresult = mycursor.fetchall()
```

```
if (mycursor.rowcount>0):
```

```
    print(" |-----  
-----|")
```

```
    print(" | empid",space("empid"),"employeenam  
space("employeenam"),"department",space("departmen  
oss_salary",space("gross_salary"),"hours_worked",space("hours_worked"),"overtime_pay",end=" ")
```

```
print(space("overtime_pay"),"income_tax",space("income_tax"),"net_salary",space('net_salary'),'inc  
rement',space('increment')," |")
```

```
    print(" |-----  
-----|")
```

```
    for x in myresult:
```

```
print(' |',x[0],space(str(x[0])),x[1],space(x[1]),x[2],space(x[2]),x[3],space(x[3]),x[4],space(str(x[4])),x[5]  
,space(str(x[5])),x[6],space(str(x[6])),x[7],space(str(x[7])),x[8],space(str(x[8])),x[9],space(str(x[9])),' |')
```

```
    print(" |-----  
-----|")
```

```
else:
```

```
print("No employee found")
```

```
elif (detail=='4'):
```

```
print("")
```

```
DEP=input("Enter Department to search:\t")
```

```
print("")
```

```
sql_select_query = "select * from emp_payroll where department = %s"
```

```
mycursor.execute(sql_select_query, (DEP, ))
```

```
myresult = mycursor.fetchall()
```

```
if (mycursor.rowcount>0):
```

```
print(" |-----  
-----|")
```

```
print(" | empid",space("empid"),"employeeename",  
space("employeeename"),"department",space("department"),"designation",space("designation"),"gr  
oss_salary",space("gross_salary"),"hours_worked",space("hours_worked"),"overtime_pay",end=" ")
```

```
print(space("overtime_pay"),"income_tax",space("income_tax"),"net_salary",space('net_salary'),'inc  
rement',space('increment')," |")
```

```
print(" |-----  
-----|")
```

```
for x in myresult:
```

```

print(' | ',x[0],space(str(x[0])),x[1],space(x[1]),x[2],space(x[2]),x[3],space(x[3]),x[4],space(str(x[4])),x[5],
space(str(x[5])),x[6],space(str(x[6])),x[7],space(str(x[7])),x[8],space(str(x[8])),x[9],space(str(x[9])),' | ')

    print(" |-----|")
-----|")

else:

    print("No employee found")

elif (detail=='5'):

    print("")

    print("""
*****
*****

*      TO SERACH GREATER THAN EQUAL TO A CERTAIN SALARY      1      *

*      TO SERACH LESSER THAN EQUAL TO A CERTAIN SALARY      2      *

*****
*****""")

    print("")

    val=input('ENTER YOUR CHOICE:\t')

    if (val=='1'):

        print("")

        SAL=int(input("Enter Salary to search:\t"))

```

```
print("")
```

```
sql_select_query = "select * from emp_payroll where gross_salary >= %s"
```

```
mycursor.execute(sql_select_query, (SAL, ))
```

```
myresult = mycursor.fetchall()
```

```
if (mycursor.rowcount>0):
```

```
    print(" |-----|")
    -----|")
```

```
        print(" | empid",space("empid"),"employeeename",
space("employeeename"),"department",space("department"),"designation",space("designation"),"gr
oss_salary",space("gross_salary"),"hours_worked",space("hours_worked"),"overtime_pay",end=" ")
```

```
print(space("overtime_pay"),"income_tax",space("income_tax"),"net_salary",space('net_salary'),'inc
rement',space('increment')," |")
```

```
    print(" |-----|")
    -----|")
```

```
for x in myresult:
```

```
print(' |',x[0],space(str(x[0])),x[1],space(x[1]),x[2],space(x[2]),x[3],space(x[3]),x[4],space(str(x[4])),x[5]
,space(str(x[5])),x[6],space(str(x[6])),x[7],space(str(x[7])),x[8],space(str(x[8])),x[9],space(str(x[9])),' |')
```

```
    print(" |-----|")
    -----|")
```

```
else:
```

```
    print("No employee found")
```



```

elif (val=='2'):

    print("")

    SAL=int(input("Enter Salary to search:\t"))

    print("")

    sql_select_query = "select * from emp_payroll where gross_salary <= %s"

    mycursor.execute(sql_select_query, (SAL, ))

    myresult = mycursor.fetchall()

    if (mycursor.rowcount>0):

        print(" |-----|")
        -----|")

        print(" | empid",space("empid"),"employeenam",
space("employeenam"),"department",space("department"),"designation",space("designation"),"gr
oss_salary",space("gross_salary"),"hours_worked",space("hours_worked"),"overtime_pay",end=" ")

print(space("overtime_pay"),"income_tax",space("income_tax"),"net_salary",space('net_salary'),'inc
rement',space('increment')," |")

        print(" |-----|")
        -----|")

        for x in myresult:

print(' |',x[0],space(str(x[0])),x[1],space(x[1]),x[2],space(x[2]),x[3],space(x[3]),x[4],space(str(x[4])),x[5]
,space(str(x[5])),x[6],space(str(x[6])),x[7],space(str(x[7])),x[8],space(str(x[8])),x[9],space(str(x[9])),' | ')

```

```
print(" |-----|")
-----|")
```

```
else:
```

```
    print("No employee found")
```

```
else:
```

```
    print('WRONG CHOICE ENTERED')
```

```
def delete_record():
```

```
    mycon = mysql.connector.connect(
```

```
        host="localhost",
```

```
        user="root",
```

```
        passwd="somerville",
```

```
        database="emp_database"
```

```
    )
```

```
    mycursor = mycon.cursor()
```

```
    print("""
```

```
*****
```

```
*      TO DELETE USING EMPLOYEE NAME      1      *
```

```
*      TO DELETE USING EMPLOYEE ID        2      *
```

```
*****")
```

```
print('')
```

```
detail=input('ENTER YOUR CHOICE:\t')
```

```
if (detail=='1'):
```

```
    print('')
```

```
    NAME=input("Enter name to delete:\t")
```

```
    print('')
```

```
    sql_select_query = "DELETE FROM emp_payroll WHERE employeename = %s"
```

```
    mycursor.execute(sql_select_query, (NAME, ))
```

```
    mycon.commit()
```

```
    print(mycursor.rowcount, "record(s) deleted from official details table")
```

```
mycon = mysql.connector.connect(
```

```
    host="localhost",
```

```
    user="root",
```

```
    passwd="somerville",
```

```
    database="emp_database"
```

```
)
```

```
mycursor = mycon.cursor()
```

```
sql_select_query = "DELETE FROM emp_personal WHERE employeename = %s"
```

```
mycursor.execute(sql_select_query, (NAME, ))
```

```
mycon.commit()
```

```
print(mycursor.rowcount, "record(s) deleted from personal details table")
```

```
elif (detail=='2'):
```

```
print("")
```

```
ID=int(input("Enter employee ID to delete:\t"))
```

```
print("")
```

```
sql_select_query = "DELETE FROM emp_payroll WHERE empid = %s"
```

```
mycursor.execute(sql_select_query, (ID, ))
```

```
mycon.commit()
```

```
print(mycursor.rowcount, "record(s) deleted from offical details table")
```

```
mycon = mysql.connector.connect(
```

```
host="localhost",  
user="root",  
passwd="somerville",  
database="emp_database"  
)
```

```
mycursor = mycon.cursor()
```

```
sql_select_query = "DELETE FROM emp_personal WHERE empid = %s"
```

```
mycursor.execute(sql_select_query, (ID, ))
```

```
mycon.commit()
```

```
print(mycursor.rowcount, "record(s) deleted from personal details table")
```

```
else:
```

```
print('WRONG ENTRY')
```

```
def update_record():
```

```
mycon = mysql.connector.connect(  
    host="localhost",  
    user="root",  
    passwd="somerville",  
    database="emp_database"
```

```
)
```

```
mycursor = mycon.cursor()
```

```
detail=int(input('Enter Employee id for updation:\t'))
```

```
sql_select_query = "select * from emp_payroll where empid = %s"
```

```
mycursor.execute(sql_select_query, (detail, ))
```

```
myresult = mycursor.fetchall()
```

```
if (len(myresult)!=0):
```

```
    print("""
```

```
*****
```

```
*   TO UPDATE DEPARTMANT OF THE EMPLOYEE           1       *
```

```
*   TO UPDATE DESIGNATION OF THE EMPLOYEE          2       *
```

```
*   TO UPDATE SALARY OF THE EMPLOYEE                3       *
```

```
*****""")
```

```
    print("")
```

```
    sdetail=input("ENTER YOUR CHOICE:\t")
```

```
    if (sdetail=='1'):
```

```
print("")
```

```
NewDept=input("Enter new Department of employee:\t")
```

```
print("")
```

```
sql_select_query = "UPDATE emp_payroll set department =%s WHERE empid = %s"
```

```
input1=(NewDept,detail)
```

```
mycursor.execute(sql_select_query,input1)
```

```
mycon.commit()
```

```
print(mycursor.rowcount, "record(s) updated")
```

```
mycursor.execute("SELECT * FROM emp_payroll")
```

```
myresult = mycursor.fetchall()
```

```
print(" |-----|")
-----|")
```

```
print(" | empid",space("empid"),"employeenam",
space("employeenam"),"department",space("department"),"designation",space("designation"),"gross_salary",space("gross_salary"),"hours_worked",space("hours_worked"),"overtime_pay",end=" ")

print(space("overtime_pay"),"income_tax",space("income_tax"),"net_salary",space('net_salary'),'increment',space('increment')," |")
```

```
print(" |-----|")
-----|")
```

```
for x in myresult:
```

```
print(' |',x[0],space(str(x[0])),x[1],space(x[1]),x[2],space(x[2]),x[3],space(x[3]),x[4],space(str(x[4])),x[5]
,space(str(x[5])),x[6],space(str(x[6])),x[7],space(str(x[7])),x[8],space(str(x[8])),x[9],space(str(x[9])),' |')
```

```
print(" |-----|")
-----|")
```

```
elif (sdetail=='2'):
```

```
print("")
```

```
NewDesig=input("Enter new Designation of employee:\t")
```

```
print("")
```

```
sql_select_query = "UPDATE emp_payroll set designation =%s WHERE empid = %s"
```

```
input1=(NewDesig,detail)
```

```
mycursor.execute(sql_select_query,input1)
```

```
mycon.commit()
```

```
print(mycursor.rowcount, "record(s) updated")
```

```
mycursor.execute("SELECT * FROM emp_payroll")
```



```
myresult = mycursor.fetchall()
```

```
print(" |-----|")
```

```
print(" | empid",space("empid"),"employeenam",
space("employeenam"),"department",space("department"),"designation",space("designation"),"gr
oss_salary",space("gross_salary"),"hours_worked",space("hours_worked"),"overtime_pay",end=" ")

print(space("overtime_pay"),"income_tax",space("income_tax"),"net_salary",space('net_salary'),'inc
rement',space('increment')," |")
```

```
print(" |-----|")
```

```
for x in myresult:
```

```
print(' |',x[0],space(str(x[0])),x[1],space(x[1]),x[2],space(x[2]),x[3],space(x[3]),x[4],space(str(x[4])),x[5]
,space(str(x[5])),x[6],space(str(x[6])),x[7],space(str(x[7])),x[8],space(str(x[8])),x[9],space(str(x[9])),' |')
```

```
print(" |-----|")
```

```
elif (sdetail=='3'):
```

```
print("")
```

```
NewSal=float(input("Enter new Salary of employee:\t"))
```

```
print("")
```

```
sql_select_query = "select * from emp_payroll where empid = %s"
```

```
mycursor.execute(sql_select_query, (detail, ))
```

```
myresult = mycursor.fetchall()
```

```
for x in myresult:
```

```
    hours=int(x[5])
```

```
    gsal=float(x[4])
```

```
dif_hour=int(hours-40)
```

```
newop=(0.05*dif_hour*int(NewSal))
```

```
newtax=(0.05*int(NewSal))
```

```
newnetsal=(NewSal+newop-newtax)
```

```
increment=(NewSal-gsal)
```

```
sql_select_query = "UPDATE emp_payroll set gross_salary =%s WHERE empid = %s"
```

```
input1=(NewSal,detail)
```

```
mycursor.execute(sql_select_query,input1)
```

```
mycon.commit()
```

```
sql_select_query = "UPDATE emp_payroll set overtime_pay =%s WHERE empid = %s"
```

```
input2=(newop,detail)
```

```
mycursor.execute(sql_select_query,input2)
```

```
mycon.commit()
```

```
sql_select_query = "UPDATE emp_payroll set income_tax =%s WHERE empid = %s"
```

```
input3=(newtax,detail)
```

```
mycursor.execute(sql_select_query,input3)
```

```
mycon.commit()
```

```
sql_select_query = "UPDATE emp_payroll set net_salary =%s WHERE empid = %s"
```

```
input4=(newnetsal,detail)
```

```
mycursor.execute(sql_select_query,input4)
```

```
mycon.commit()
```

```
sql_select_query = "UPDATE emp_payroll set increment =%s WHERE empid = %s"
```

```
input5=(increment,detail)
```

```
mycursor.execute(sql_select_query,input5)
```

```
mycon.commit()
```

```
print(mycursor.rowcount, "record(s) updated")
```

```
mycursor.execute("SELECT * FROM emp_payroll")
```

```
myresult = mycursor.fetchall()
```

```
print(" |-----  
-----|")
```

```
print(" | empid",space("empid"),"employee name",  
space("employee name"),"department",space("department"),"designation",space("designation"),"gr  
oss_salary",space("gross_salary"),"hours_worked",space("hours_worked"),"overtime_pay",end=" ")  
  
print(space("overtime_pay"),"income_tax",space("income_tax"),"net_salary",space('net_salary'),'inc  
rement',space('increment')," |")
```

```
print(" |-----  
-----|")
```

```
for x in myresult:
```

```
print(' |',x[0],space(str(x[0])),x[1],space(x[1]),x[2],space(x[2]),x[3],space(x[3]),x[4],space(str(x[4])),x[5]  
,space(str(x[5])),x[6],space(str(x[6])),x[7],space(str(x[7])),x[8],space(str(x[8])),x[9],space(str(x[9])),' |')
```

```
print(" |-----  
-----|")
```

```
else:
```

```
print("")
```

```
print('WRONG ENTRY')
```

```
elif (len(myresult)==0):
```

```
print("")
```

```
print('NO EMPLOYEE WITH SUCH EMPLOYEE ID AS PER OFFICIAL DATA')
```

#-----

#FOR TABLE 2:

```
def create_table_personal():
```

```
    mycon = mysql.connector.connect(
```

```
        host="localhost",
```

```
        user="root",
```

```
        passwd="somerville",
```

```
        database="emp_database"
```

```
    )
```

```
    mycursor = mycon.cursor()
```

```
    mycursor.execute("CREATE TABLE IF NOT EXISTS emp_personal(empid int(5) primary key  
auto_increment,employeenname varchar(20),gender varchar(7),address varchar(25),mobile_number  
varchar(10),Qualification varchar(10))")
```

```
    print("Tables under emp_database: ")
```

```
    mycursor.execute("SHOW TABLES")
```

```
    for x in mycursor:
```

```
        print(x)
```

```
def add_new_record_personal_admin():
```

```
    mycon = mysql.connector.connect(
```

```
host="localhost",

user="root",

passwd="somerville",

database="emp_database"

)

mycursor = mycon.cursor()


empid=input('ENTER EMPLOYEE ID:\t')


sql_select_query = "select * from emp_payroll where empid = %s"


mycursor.execute(sql_select_query, (empid, ))


myresult = mycursor.fetchall()


if len(myresult)!=0:


    for x in myresult:


        employeename=x[1]


        print('THE NAME OF EMPLOYEE AS PER OFFICAL DATA:\t',employeename)


gender=input("ENTER GENDER:\t")


address=input("ENTER ADDRESS:\t")
```

```
mobile_number=int(input("ENTER MOBILE NUMBER:\t"))
```

```
qualification=input("ENTER QUALIFICATION\t")
```

```
sql = "INSERT INTO  
emp_personal(empid,employeename,gender,address,mobile_number,qualification)  
VALUES({}, '{}', '{}', '{}', '{}', '{}')".format(empid,employeename,gender,address,mobile_number,qualifica  
tion)
```

```
mycursor.execute(sql)
```

```
mycon.commit()
```

```
print(mycursor.rowcount, "record inserted.")
```

```
elif len(myresult)==0:
```

```
print('NO EMPLOYEE WITH SUCH EMPLOYEE ID AS PER OFFICAL DATA')
```

```
def add_new_record_personal():
```

```
mycon = mysql.connector.connect(
```

```
host="localhost",
```

```
user="root",
```

```
passwd="somerville",
```

```
database="emp_database"
```

```
)
```

```
mycursor = mycon.cursor()
```

empid=ID

sql_select_query = "select * from emp_payroll where empid = %s"

mycursor.execute(sql_select_query, (empid,))

myresult = mycursor.fetchall()

if len(myresult)!=0:

for x in myresult:

 employee_name=x[1]

 print('THE NAME OF EMPLOYEE AS PER OFFICAL DATA:',employee_name)

gender=input("ENTER GENDER:\t")

address=input("ENTER ADDRESS:\t")

mobile_number=input("ENTER MOBILE NUMBER:\t")

qualification=input("ENTER QUALIFICATION\t")

sql = "INSERT INTO

emp_personal(empid,employee_name,gender,address,mobile_number,qualification)


```
VALUES({}, '{}', '{}', '{}', '{}', '{}')".format(empid, employeename, gender, address, mobile_number, qualification)
```

```
mycursor.execute(sql)
```

```
mycon.commit()
```

```
print(mycursor.rowcount, "record inserted.")
```

```
elif len(myresult)==0:
```

```
print("")
```

```
print('NO EMPLOYEE WITH SUCH EMPLOYEE ID AS PER OFFICAL DATA')
```

```
def search_record_personal():
```

```
mycon = mysql.connector.connect(
```

```
    host="localhost",
```

```
    user="root",
```

```
    passwd="somerville",
```

```
    database="emp_database"
```

```
)
```

```
mycursor = mycon.cursor()
```

global ID

```
sql_select_query = "select * from emp_personal where empid = %s"
```

```
mycursor.execute(sql_select_query, (ID, ))
```

```
myresult = mycursor.fetchall()
```

```
if len(myresult)!=0:
```

```
    sql_select_query = "select * from emp_personal where empid = %s"
```

```
    mycursor.execute(sql_select_query, (ID, ))
```

```
    myresult = mycursor.fetchall()
```

```
    print("|-----|")
```

```
    print("| empid",space("empid"),"employeenam",  
space("employeenam"),"gender",space("gender"),"address",space("address"),"mobile_number",sp  
ace("mobile_number"),"qualification",space("qualification")," |")
```

```
    print("|-----|")
```

```
    for x in myresult:
```

```
print(' |',x[0],space(str(x[0])),x[1],space(x[1]),x[2],space(x[2]),x[3],space(x[3]),x[4],space(str(x[4])),x[5],space(str(x[5])),' |')
```

```
print(" |-----|")
```

```
elif len(myresult)==0:
```

```
print("")
```

```
print('NO EMPLOYEE WITH SUCH EMPLOYEE ID AS PER OFFICAL DATA')
```

```
def update_record_personal():
```

```
mycon = mysql.connector.connect(
```

```
    host="localhost",
```

```
    user="root",
```

```
    passwd="somerville",
```

```
    database="emp_database"
```

```
)
```

```
mycursor = mycon.cursor()
```

```
global ID
```

```
sql_select_query = "select * from emp_personal where empid = %s"
```

```
mycursor.execute(sql_select_query, (ID, ))
```

```
myresult = mycursor.fetchall()
```

```
if len(myresult)!=0:
```

```
    print("")
```

```
    *****
```

```
    *   TO UPDATE ADDRESS OF THE EMPLOYEE           1           *
```

```
    *   TO UPDATE MOBILE NUMBER OF THE EMPLOYEE      2           *
```

```
    *   TO UPDATE QUALIFICATIONS OF THE EMPLOYEE     3           *
```

```
    *****")
```

```
    print("")
```

```
    sdetail_personal=input("ENTER YOUR CHOICE:\t")
```

```
    if (sdetail_personal=='1'):
```

```
        print("")
```

```
        NewAdd=input("Enter your new Address:\t")
```

```
        print("")
```

```
sql_select_query = "UPDATE emp_personal set address =%s WHERE empid = %s"
```

```
input5=(NewAdd,ID)
```

```
mycursor.execute(sql_select_query,input5)
```

```
mycon.commit()
```

```
print(mycursor.rowcount, "record(s) updated")
```

```
sql_select_query="SELECT * FROM emp_personal where empid= %s"
```

```
mycursor.execute(sql_select_query, (ID, ))
```

```
myresult = mycursor.fetchall()
```

```
print(" |-----|")
```

```
print(" | empid",space("empid"),"employeeename",  
space("employeeename"),"gender",space("gender"),"address",space("address"),"mobile_number",sp  
ace("mobile_number"),"qualification",space("qualification")," |")
```

```
print(" |-----|")
```

```
for x in myresult:
```

```
print(' |',x[0],space(str(x[0])),x[1],space(x[1]),x[2],space(x[2]),x[3],space(x[3]),x[4],space(str(x[4])),x[5]  
,space(str(x[5])),' |')
```

```
print(" |-----|")
```

```
elif (sdetail_personal=='2'):
```

```
    print("")
```

```
    NewNumber=input("Enter your new Mobile Number:\t")
```

```
    print("")
```

```
    sql_select_query = "UPDATE emp_personal set mobile_number =%s WHERE empid = %s"
```

```
    input6=(NewNumber,ID)
```

```
    mycursor.execute(sql_select_query,input6)
```

```
    mycon.commit()
```

```
    print(mycursor.rowcount, "record(s) updated")
```

```
    sql_select_query="SELECT * FROM emp_personal where empid= %s"
```

```
    mycursor.execute(sql_select_query, (ID, ))
```

```
    myresult = mycursor.fetchall()
```

```
    print(" |-----|")
```

```
    print(" | empid",space("empid"),"employeenam",
```

```
space("employeenam"),"gender",space("gender"),"address",space("address"),"mobile_number",space("mobile_number"),"qualification",space("qualification")," |")
```

```
print(" |-----|")
```

```
for x in myresult:
```

```
print(' |',x[0],space(str(x[0])),x[1],space(x[1]),x[2],space(x[2]),x[3],space(x[3]),x[4],space(str(x[4])),x[5]  
,space(str(x[5])),' |')
```

```
print(" |-----|")
```

```
elif (sdetail_personal=='3'):
```

```
print("")
```

```
NewQual=input("Enter your new Qualification:\t")
```

```
print("")
```

```
sql_select_query = "UPDATE emp_personal set qualification =%s WHERE empid = %s"
```

```
input7=(NewQual,ID)
```

```
mycursor.execute(sql_select_query,input7)
```

```
mycon.commit()
```

```
print(mycursor.rowcount, "record(s) updated")
```

```
sql_select_query="SELECT * FROM emp_personal where empid= %s"
```

```
mycursor.execute(sql_select_query, (ID, ))
```

```
myresult = mycursor.fetchall()
```

```
print(" |-----|")
```

```
print(" | empid",space("empid"),"employeenam",  
space("employeenam"),"gender",space("gender"),"address",space("address"),"mobile_number",sp  
ace("mobile_number"),"qualification",space("qualification")," |")
```

```
print(" |-----|")
```

```
for x in myresult:
```

```
print(' |',x[0],space(str(x[0])),x[1],space(x[1]),x[2],space(x[2]),x[3],space(x[3]),x[4],space(str(x[4])),x[5]  
,space(str(x[5])),' |')
```

```
print(" |-----|")
```

```
else:
```

```
print("")
```

```
print('WRONG ENTRY PLS CHECK AGAIN')
```

```
else:
```

```
print("")
```

```
print("NO EMPLOYEE WITH SUCH EMPLOYEE ID AS PER OFFICAL DATA")
```

```
def display_record_personal():
```

```
mycon = mysql.connector.connect(
```



```

host="localhost",

user="root",

passwd="somerville",

database="emp_database"

)


mycursor = mycon.cursor()


mycursor.execute("SELECT * FROM emp_personal")


myresult = mycursor.fetchall()

print("|-----|")

print("| empid",space("empid"),"employeeename",
space("employeeename"),"gender",space("gender"),"address",space("address"),"mobile_number",sp
ace("mobile_number"),"qualification",space("qualification")," |")


print("|-----|")


for x in myresult:

print(' |',x[0],space(str(x[0])),x[1],space(x[1]),x[2],space(x[2]),x[3],space(x[3]),x[4],space(str(x[4])),x[5]
,space(str(x[5])),' |')

print("|-----|")


def delete_record_personal():

```

```
mycon = mysql.connector.connect(  
    host="localhost",  
    user="root",  
    passwd="somerville",  
    database="emp_database"  
)
```

```
mycursor = mycon.cursor()
```

```
global ID
```

```
sql_select_query = "DELETE FROM emp_personal WHERE empid = %s"
```

```
mycursor.execute(sql_select_query, (ID, ))
```

```
mycon.commit()
```

```
print("")
```

```
print(mycursor.rowcount, "record(s) deleted")
```

```
def payroll_slip_data_from_personal_table():
```

```
    mycon = mysql.connector.connect(  
        host="localhost",  
        user="root",  
        passwd="somerville",
```

```
database="emp_database"

)

mycursor = mycon.cursor()

global ID

sql_select_query = "select * from emp_personal where empid = %s"

mycursor.execute(sql_select_query, (ID, ))

myresult = mycursor.fetchall()

if len(myresult)!=0:

    sql_select_query = "select * from emp_personal where empid = %s"

    mycursor.execute(sql_select_query, (ID, ))

    myresult = mycursor.fetchall()

    for x in myresult:

        print('\t\t-----')
        print('\t\tEMPLOYEE PAY SLIP')
        print('\t\t-----')
        print('\t\tID      : ',str(x[0]) )
        print("")
        print('\t\tNAME      : ',x[1] )
```

```
print("")  
  
print('\t\tGENDER      : ',x[2] )  
  
print("")  
  
print('\t\tADDRESS      : ',x[3] )  
  
print("")  
  
print('\t\tMOBILE NUMBER : ',str(x[4]) )  
  
print("")  
  
print('\t\tQUALIFICATION : ',x[5] )
```

```
elif len(myresult)==0:
```

```
    print("")  
  
    print('PERSONAL/PROFESSIONAL DETAILS OF THIS EMPLOYEE NOT FOUND')
```

```
def payroll_slip():
```

```
    mycon = mysql.connector.connect(  
  
        host="localhost",  
  
        user="root",  
  
        passwd="somerville",  
  
        database="emp_database"  
  
    )
```

```
    mycursor = mycon.cursor()
```

```
    global ID
```

```
    sql_select_query = "select * from emp_personal where empid = %s"
```

```
mycursor.execute(sql_select_query, (ID, ))
```

```
myresult = mycursor.fetchall()
```

```
if len(myresult)!=0:
```

```
    sql_select_query = "select * from emp_payroll where empid = %s"
```

```
    mycursor.execute(sql_select_query,(ID, ))
```

```
    myresult = mycursor.fetchall()
```

```
    emp_underperforming_quotes = ['Life's like a movie, write your own ending. Keep believing,  
keep pretending. —Jim Hensen', 'So many things are possible just as long as you don't know they're  
impossible. —Norton Juster',
```

```
        'The moment you doubt whether you can fly, you cease forever to be able to do  
it.', 'Opportunity is missed by most people because it is dressed in overalls and looks like work. —  
Thomas Edison']
```

```
    emp_good_performance_quotes = ['Your dedication is imperative for the growth of our  
company. Thank you for your efforts.',
```

```
        'It's the vision of employees like you who turn desired plans into success. We  
value you and your work to the moon and back.',
```

```
        'An employee's hard work is the fuel to great company culture. Thanks for  
adding extra miles to ours.',
```

```
        'The service you provided exceeded all expectations. I would also like to add  
how much you mean to our company. Thank you, and keep up with the good work.']
```

```
for x in myresult:
```

```
    print("")
```

```
print('\t\tDEPARTMENT   : ',x[2] )
print("")
print('\t\tDESIGNATION   : ',x[3] )
print("")
print('\t\tGROSS SALARY   : ',x[4] )
print("")
print('\t\tHOURS WORKED   : ',x[5],"Hours" )
print("")
print('\t\tOVERTIME PAY   : ',x[6] )
print("")
print('\t\tINCOME TAX     : ',x[7] )
print("")
print('\t\tNET SALARY     : ',x[8] )
print("")
print('\t\tINCREMENT      : ',x[9] )
print('\n\n\n')
```

```
print('MOTIVATIONAL LINES FOR YOU:\n')
```

```
if (x[5]>=170):
```

```
    print('\t',random.choice(emp_good_performance_quotes))
```

```
    print('\n')
```

```
else:
```

```
    print('\t',random.choice(emp_underperforming_quotes))
```

```
    print('\n')
```

```
elif len(myresult)==0:
```

```
    print("")
```

```
print('PERSONAL/PROFESSIONAL DETAILS OF THIS EMPLOYEE NOT FOUND')
```

```
def emp_recomendation():
```

```
    mycon = mysql.connector.connect(
```

```
        host="localhost",
```

```
        user="root",
```

```
        passwd="somerville",
```

```
        database="emp_database"
```

```
    )
```

```
    mycursor = mycon.cursor()
```

```
    global ID
```

```
    sql_select_query = "select * from emp_payroll where empid = %s"
```

```
    mycursor.execute(sql_select_query, (ID, ))
```

```
    myresult = mycursor.fetchall()
```

```
    if len(myresult)!=0:
```

```
        for x in myresult:
```

```
            hour=x[5]
```

```
            diff=int(hour-40)
```

```
            if (diff>=20):
```

```
print("")
```

```
print('*****EMPLOYEE GRADE-  
A1*****')
```

```
print("")
```

```
print('YOUR PERFORMANCE HAS BEEN REALLY AMAZING. YOU ARE ONE OF THE STAR  
EMPLOYEE OF THE COMPANY. KEEP UP THE GOOD WORK.')
```

```
print("")
```

```
elif (diff<=20 and diff>=0):
```

```
print("")
```

```
print('*****EMPLOYEE GRADE-  
A2*****')
```

```
print("")
```

```
print('YOUR WORK IS HIGHLY APPRECIATED. YOU ARE IN THE GOOD BOOKS OF YOUR BOSSES.  
KEEP UP THE GOOD WORK.')
```

```
print("")
```

```
elif (diff<=0 and diff>=-10):
```

```
print("")
```

```
print('*****EMPLOYEE GRADE-  
B1*****')
```

```
print("")
```

```
print('YOUR PERFORMANCE HAS BEEN GOOD. CONTINUE TO YOUR HARDWORK AND AIM  
FOR HIGHER GOALS.')
```

```
print("")
```

```
elif (diff<=-20):
```

```
print("")
```

```
print('*****EMPLOYEE GRADE-  
C1*****')
```



```
print("")
```

```
print('YOUR PERFORMANCE HAS BEEN EXTREMELY AVERAGE. YOU SHOULD INCREASE YOUR  
EFFORT AND FOCUS MORE ON YOUR WORK.')
```

```
print("")
```

```
elif len(myresult)==0:
```

```
print("")
```

```
print('PERSONAL/PROFESSIONAL DETAILS OF THIS EMPLOYEE NOT FOUND')
```

```
#-----  
-----#
```

```
#_main_
```

```
x=check_connection()
```

```
if x==1:
```

```
    create_database()
```

```
    create_table()
```

```
    create_table_personal()
```

```
else:
```

```
    print("Kindly check connection")
```

```
    print("")
```

```
print("-----")
```

```
print("  
|=====|")
```

```

print("                                |===== Welcome To Employee Payroll Management
System =====|")

print("
|=====|")

print("                                -----")

print("")

print("")

print("")

print("""
*****

*   WANT TO WORK AS ADMIN, THEN ENTER   1       *

*****

*   WANT TO WORK AS EMPLOYEE, THEN ENTER 2       *

*****""")

print("")

user=input("ENTER YOUR CHOICE:\t")

if (user=="1"):

    c=0

    while(c<=2):

```

```
print("")
```

```

print("*****")

    print("")

    print("*\t\t4> Delete Offical Details    ||    4    *")

    print("")

print("*****")

    print("")

    print("*\t\t5> Update Offical Details    ||    5    *")

    print("")

print("*****")

    print("")

    print("*\t\t6> Add Personal Details    ||    6    *")

    print("")

print("*****")

    print("")

    print("*\t\t7> View Personal Details    ||    7    *")

    print("")

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

ch=int(input("enter your choice\t"))

print("")

if ch==1:

    add_new_record()

elif ch==2:

    display_all_record()

```

```
elif ch==3:
```

search_record()

```
elif ch==4:
```

delete_record()

```
elif ch==5:
```

update_record()

```
elif ch==6:
```

```
add_new_record_personal_admin()
```

```
elif ch==7:
```

```
display_record_personal()
```

else:

```
print("Wrong Choice, Please enter values between 1-7")
```

```
print("")
```

```
ans=input("Wish to continue\t")
```

```
if (ans!='y' or ans!='Y'):
```

$c=4$

```
print("\t\t\t\t\tTHANK YOU FOR USING EMPLOYEE DATABASE MANAGEMENT SYSTEM")
```

```
elif(password=='ADMIN'):
```

```
print('\nWRONG PASSWORD ENTERED!!!')
```

```
print('\nCAUTION:CAPS LOCK IS ON')
```

C=C+1

```
elif(password!='admin' and password!='ADMIN'):
```

```
print('\nWRONG PASSWORD ENTERED!!!')
```

C=C+1

```
if (c==3):
```

```
    print('INCORRECT PASSWORD LIMIT REACHED')
```

```
    print('ACCESS DENIED!!!')
```

```
elif (user=="2"):
```

```
    a='y'
```

```
    while(a=='y' or a=='Y'):
```

```
        print("")
```

```
        ID=int(input('Enter Employee ID:\t'))
```

```
        print("")
```

```
        print("")
```

```
print("*****")
```

```
    print("*\t\tEMPLOYEE PERSONAL DETAILS MANAGEMENT SYSTEM\t\t*")
```

```
print("*****")
```

```
    print("")
```

```
    print("*\t\t1> Add Personal Details\t\t| |\t\t1\t\t*")
```

```
    print("")
```

```
print("*****")
```

```
    print("")
```

```
    print("*\t\t2> Search Personal Details\t\t| |\t\t2\t\t*")
```

```

print("")

print("*****")

print("")

print("*\t\t3> Delete Personal Details    ||    3    *")

print("")

print("*****")

print("")

print("*\t\t4> Update Personal Details    ||    4    *")

print("")

print("*****")

print("")

print("*\t\t5> Print Payslip                ||    5    *")

print("")

print("*****")

print("")

print("*\t\t6> Remarks From Admin          ||    6    *")

print("")

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

a=int(input('Enter your Choice:\t'))

if (a==1):

    add_new_record_personal()

```

```
elif (a==2):
```

```
search_record_personal()
```

```
elif (a==3):
```

```
delete_record_personal()
```

```
elif (a==4):
```

```
update_record_personal()
```

```
elif (a==5):
```

payroll_slip_data_from_personal_table(), payroll_slip()

```
elif(a==6):
```

emp_recomendation()

else:

```
print('WRONG ENTRY')
```

```
print('')
```

```
a=input('Wish to continue:\t')
```

```
print('\t\t\t\t\tTHANK YOU FOR USING EMPLOYEE DATABASE MANAGEMENT SYSTEM')
```

```
else:
```

```
print('WRONG OPTION ENTERED')
```

SOURCE CODE:



The user needs to double click on this zip folder icon to view the contents and then he can extract the python file from this compressed zip folder and thereby run it using IDLE

OUTPUT REPORTS

WELCOME SCREEN:

```
-----  
|=====|  
|===== Welcome To Employee Payroll Management System =====|  
|=====|  
|-----|
```

```
*****
```

```
*   WANT TO WORK AS ADMIN, THEN ENTER      1           *
```

```
*****
```

```
*   WANT TO WORK AS EMPLOYEE, THEN ENTER    2           *
```

```
*****
```

```
ENTER YOUR CHOICE:      1
```

```
ENTER PASSWORD: admin
```

WORKING AS ADMIN:

```
*****
*                               EMPLOYEE PAYROLL MANAGEMENT SYSTEM                               *
*****

1>  Add Offical Details          ||          1          *

*****

2>  View Offical Details         ||          2          *

*****

3>  Search Official Details      ||          3          *

*****

4>  Delete Offical Details       ||          4          *

*****

5>  Update Offical Details       ||          5          *

*****

6>  Add Personal Details         ||          6          *

*****

7>  View Personal Details        ||          7          *

*****
```

ADDING OFFICIAL DETAILS:

```
enter your choice      1

ENTER EMPLOYEE ID:      1
ENTER EMPLOYEE NAME:    Ayush
ENTER DEPARTMENT OF EMPLOYEE:    Sales
ENTER DESIGNATION OF EMPLOYEE    SAles Manager
ENTER GROSS SALARY OF THE EMPLOYEE    200000
ENTER TOTAL HOURS WORKED      42

1 record inserted.
```

VIEWING OFFICIAL DETAILS:

```
enter your choice      2
```

empid	employeename	department	designation	gross_salary	hours_worked	overtime_pay	income_tax	net_salary	increment
1	Ayush	Sales	SAles Manager	200000.0	42	20000.0	10000.0	210000.0	0.0
2	Kushagr	R&D	Secretary	350000.0	40	0.0	17500.0	332500.0	0.0

SEARCHING OFFICIAL DETAILS:

I. SEARCHING USING EMPLOYEE NAME:

```
*****
*      TO SERACH USING EMPLOYEE NAME      1      *
*      TO SEARCH USING EMPLOYEE ID        2      *
*      TO SEARCH USING DESIGNATION        3      *
*      TO SEARCH USING DEPARTMENT        4      *
*      TO SEARCH USING GROSS SALARY       5      *
*****
```

ENTER YOUR CHOICE: 1

Enter Employee name to search: aa

empid	employeename	department	designation	gross_salary	hours_worked	overtime_pay	income_tax	net_salary	increment
150	aa	aa	aa	150.0	40	0.0	7.5	142.5	0.0

II. SEARCHING USING EMPLOYEE ID:

```
*****
*      TO SERACH USING EMPLOYEE NAME      1      *
*      TO SEARCH USING EMPLOYEE ID        2      *
*      TO SEARCH USING DESIGNATION        3      *
*      TO SEARCH USING DEPARTMENT        4      *
*      TO SEARCH USING GROSS SALARY       5      *
*****
```

ENTER YOUR CHOICE: 2

Enter Employee id to search: 150

empid	employeename	department	designation	gross_salary	hours_worked	overtime_pay	income_tax	net_salary	increment
150	aa	aa	aa	150.0	40	0.0	7.5	142.5	0.0

III. SEARCHING USING DESIGNATION:

```
*****
*      TO SERACH USING EMPLOYEE NAME      1      *
*      TO SEARCH USING EMPLOYEE ID        2      *
*      TO SEARCH USING DESIGNATION        3      *
*      TO SEARCH USING DEPARTMENT        4      *
*      TO SEARCH USING GROSS SALARY       5      *
*****
```

ENTER YOUR CHOICE: 3

Enter Designation to search: engineer

empid	employeeename	department	designation	gross_salary	hours_worked	overtime_pay	income_tax	net_salary	increment
109	vimal	construction	engineer	900.0	56	720.0	45.0	1575.0	0.0
200	sanajy	IT	engineer	5000.0	45	1250.0	250.0	6000.0	0.0

IV. SEARCHING USING DEPARTMENT:

```
*****
*      TO SERACH USING EMPLOYEE NAME      1      *
*      TO SEARCH USING EMPLOYEE ID        2      *
*      TO SEARCH USING DESIGNATION        3      *
*      TO SEARCH USING DEPARTMENT        4      *
*      TO SEARCH USING GROSS SALARY       5      *
*****
```

ENTER YOUR CHOICE: 4

Enter Department to search: finance

empid	employeeename	department	designation	gross_salary	hours_worked	overtime_pay	income_tax	net_salary	increment
102	arun	finance	manager	1500.0	45	375.0	75.0	1800.0	0.0
163	priya	finance	assistant	655.0	45	163.75	32.75	786.0	0.0

V. SEARCHING USING GROSS SALARY (GREATER THAN):

```
*****
*      TO SERACH USING EMPLOYEE NAME      1      *
*      TO SEARCH USING EMPLOYEE ID        2      *
*      TO SEARCH USING DESIGNATION        3      *
*      TO SEARCH USING DEPARTMENT        4      *
*      TO SEARCH USING GROSS SALARY       5      *
*****
```

ENTER YOUR CHOICE: 5

```
*****
*      TO SERACH GREATER THAN EQUAL TO A CERTAIN SALARY      1      *
*      TO SERACH LESSER THAN EQUAL TO A CERTAIN SALARY      2      *
*****
```

ENTER YOUR CHOICE: 1

Enter Salary to search: 1000

empid	employeeename	department	designation	gross_salary	hours_worked	overtime_pay	income_tax	net_salary	increment
102	arun	finance	manager	1500.0	45	375.0	75.0	1800.0	0.0
198	thomas	construction	HOD	9000.0	50	4500.0	450.0	13050.0	0.0
200	sanajy	IT	engineer	5000.0	45	1250.0	250.0	6000.0	0.0

VI. SEARCHING USING GROSS SALARY (LESSER THAN):

```
*****
*          TO SERACH USING EMPLOYEE NAME          1          *
*          TO SEARCH USING EMPLOYEE ID            2          *
*          TO SEARCH USING DESIGNATION            3          *
*          TO SEARCH USING DEPARTMENT            4          *
*          TO SEARCH USING GROSS SALARY           5          *
*****
```

ENTER YOUR CHOICE: 5

```
*****
*          TO SERACH GREATER THAN EQUAL TO A CERTAIN SALARY 1          *
*          TO SERACH LESSER THAN EQUAL TO A CERTAIN SALARY 2          *
*****
```

ENTER YOUR CHOICE: 2

Enter Salary to search: 1500

empid	employeenam	department	designation	gross_salary	hours_worked	overtime_pay	income_tax	net_salary	increment
102	arun	finance	manager	1500.0	45	375.0	75.0	1800.0	0.0
109	vimal	construction	engineer	900.0	56	720.0	45.0	1575.0	0.0
150	aa	aa	aa	150.0	40	0.0	7.5	142.5	0.0
163	priya	finance	assistant	655.0	45	163.75	32.75	786.0	0.0

DELETING OFFICIAL DETAILS:

I. DELETING USING EMPLOYEE NAME:

```
*****
*          TO DELETE USING EMPLOYEE NAME          1          *
*          TO DELETE USING EMPLOYEE ID            2          *
*****
```

ENTER YOUR CHOICE: 1

Enter name to delete: vimal

1 record(s) deleted from offical details table
0 record(s) deleted from personal details table

II. DELETING USING EMPLOYEE ID:

```
*****

*          TO DELETE USING EMPLOYEE NAME          1          *
*          TO DELETE USING EMPLOYEE ID           2          *

*****

ENTER YOUR CHOICE:      2

Enter employee ID to delete:    200

1 record(s) deleted from official details table
0 record(s) deleted from personal details table
```

UPDATING OFFICIAL DETAILS

I. UPDATING THE DEPARTMENT:

```
Enter Employee id for updation: 1

*****

*          TO UPDATE DEPARTMANT OF THE EMPLOYEE      1          *
*          TO UPDATE DESIGNATION OF THE EMPLOYEE     2          *
*          TO UPDATE SALARY OF THE EMPLOYEE          3          *

*****

ENTER YOUR CHOICE:      1

Enter new Department of employee:      Management

1 record(s) updated

-----
| empid      employeename      department      designation      gross_salary      hours_worked      overtime_pay      income_tax      net_salary      increment
|-----|
| 1          Akshat            Management    Sales Manager    200000.0          45                50000.0          10000.0          240000.0          0.0
|-----|
```

II. UPDATING DESIGNATION:

Enter Employee id for updation: 1

```
*****
*      TO UPDATE DEPARTMANT OF THE EMPLOYEE      1      *
*      TO UPDATE DESIGNATION OF THE EMPLOYEE     2      *
*      TO UPDATE SALARY OF THE EMPLOYEE          3      *
*****
```

ENTER YOUR CHOICE: 2

Enter new Designation of employee: Management Head

1 record(s) updated

empid	employeename	department	designation	gross_salary	hours_worked	overtime_pay	income_tax	net_salary	increment
1	Akshat	Management	Management Head	200000.0	45	50000.0	10000.0	240000.0	0.0

III. UPDATING SALARY AND VIEWING THE INCREMENT:

ENTER YOUR CHOICE: 3

Enter new Salary of employee: 265000

1 record(s) updated

empid	employeename	department	designation	gross_salary	hours_worked	overtime_pay	income_tax	net_salary	increment
1	Akshat	Management	Management Head	265000.0	45	66250.0	13250.0	318000.0	65000.0

ADDING PERSONAL DETAILS:

enter your choice 6

ENTER EMPLOYEE ID: 1

THE NAME OF EMPLOYEE AS PER OFFICAL DATA: Ayush

ENTER GENDER: Male

ENTER ADDRESS: Sector-76, Noida

ENTER MOBILE NUMBER: 9365468222

ENTER QUALIFICATION M.Ba

1 record inserted.

VIEWING PERSONAL DETAILS:

enter your choice 7

empid	employeename	gender	address	mobile_number	qualification
1	Ayush	Male	Sector-76, Noida	9365468222	M.Ba

WORKING AS EMPLOYEE:

|

* WANT TO WORK AS ADMIN, THEN ENTER ADMIN *

* WANT TO WORK AS EMPLOYEE, THEN ENTER EMPLOYEE *

ENTER YOUR CHOICE: employee

Enter Employee ID: 101

* EMPLOYEE PERSONAL DETAILS MANAGEMENT SYSTEM *

* 1> Add Personal Details || 1 *

* 2> Search Personal Details || 2 *

* 3> Delete Personal Details || 3 *

* 4> Update Personal Details || 4 *

* 5> Print Payslip || 5 *

* 6> Remarks From Admin || 6 *

ADDING PERSONAL DETAILS:

Enter your Choice: 1
THE NAME OF EMPLOYEE AS PER OFFICAL DATA: ranjan
ENTER GENDER: male
ENTER ADDRESS: delhi
ENTER MOBILE NUMBER: 9654345687
ENTER QUALIFICATION MS
1 record inserted.

Wish to continue: |

VIEWING PERSONAL DETAILS:

Enter your Choice: 2

empid	employeename	gender	address	mobile_number	qualification
104	ranjan	male	delhi	9654345687	MS

Wish to continue: |

DELETING PERSONAL DETAILS:

Wish to continue: Y

Enter Employee ID: 104

```
*****
*                EMPLOYEE PERSONAL DETAILS MANAGEMENT SYSTEM                *
*****
```

```
*          1>  Add Personal Details          ||          1          *
```

```
*****
```

```
*          2>  Search Personal Details        ||          2          *
```

```
*****
```

```
*          3>  Delete Personal Details        ||          3          *
```

```
*****
```

```
*          4>  Update Personal Details        ||          4          *
```

```
*****
```

```
*          5>  Print Payslip                  ||          5          *
```

```
*****
```

```
*          6>  Remarks From Admin             ||          6          *
```

```
*****
```

Enter your Choice: 3

1 record(s) deleted

Wish to continue: |

UPDATING PERSONAL DETAILS:

I. UPDATING THE ADDRESS:

```
*****
*      TO UPDATE ADDRESS OF THE EMPLOYEE          1      *
*      TO UPDATE MOBILE NUMBER OF THE EMPLOYEE    2      *
*      TO UPDATE QUALIFICATIONS OF THE EMPLOYEE    3      *
*****
```

ENTER YOUR CHOICE): 1
Enter your new Address: Sector 40, Noida

1 record(s) updated

empid	employeename	gender	address	mobile_number	qualification
1	Akshat	Male	Sector 40, Noida	9876543234	M.BA

II. UPDATING THE MOBILE NUMBER:

Enter your Choice: 4

```
*****
*      TO UPDATE ADDRESS OF THE EMPLOYEE          1      *
*      TO UPDATE MOBILE NUMBER OF THE EMPLOYEE    2      *
*      TO UPDATE QUALIFICATIONS OF THE EMPLOYEE    3      *
*****
```

ENTER YOUR CHOICE): 2
Enter your new Mobile Number: 8237645655

1 record(s) updated

empid	employeename	gender	address	mobile_number	qualification
1	Akshat	Male	Sector 40, Noida	8237645655	M.BA

III. UPDATING THE QUALIFICATION:

```
Enter your Choice:      4

*****
*      TO UPDATE ADDRESS OF THE EMPLOYEE      1      *
*      TO UPDATE MOBILE NUMBER OF THE EMPLOYEE  2      *
*      TO UPDATE QUALIFICATIONS OF THE EMPLOYEE  3      *
*****

ENTER YOUR CHOICE):      3

Enter your new Qualification:   B.Sc

1 record(s) updated
|-----|
| empid      employeename      gender      address      mobile_number      qualification      |
|-----|
| 1          Akshat            Male       Sector 40, Noida  8237645655        B.Sc              |
|-----|
```

PRINTING PAYSLIP:

```
Enter your Choice:      5

-----
EMPLOYEE PAY SLIP
-----
ID           :   1
NAME          :  Akshat
GENDER        :   Male
ADDRESS       :  Sector 40, Noida
MOBILE NUMBER :  8237645655
QUALIFICATION :   B.Sc
DEPARTMENT    :  Management
DESIGNATION   :  Management Head
GROSS SALARY  :  265000.0
HOURS WORKED  :   45 Hours
OVERTIME PAY  :   66250.0
INCOME TAX    :   13250.0
NET SALARY    :  318000.0
INCREMENT     :   65000.0
```

MOTIVATIONAL LINES FOR YOU:

So many things are possible just as long as you don't know they're impossible. -Norton Juster

SHOWING REMARKS FROM ADMIN:

Enter Employee ID: 1

```
*****
*                EMPLOYEE PERSONAL DETAILS MANAGEMENT SYSTEM                *
*****

*          1>  Add Personal Details          ||          1          *
*****

*          2>  Search Personal Details        ||          2          *
*****

*          3>  Delete Personal Details        ||          3          *
*****

*          4>  Update Personal Details        ||          4          *
*****

*          5>  Print Payslip                  ||          5          *
*****

*          6>  Remarks From Admin             ||          6          *
*****
```

Enter your Choice: 6

*****EMPLOYEE GRADE-A2*****
YOUR WORK IS HIGHLY APPRECIATED. YOU ARE IN THE GOOD BOOKS OF YOUR BOSSES. KEEP UP THE GOOD WORK.

LIMITATIONS OF THE PROJECT:

1. This program does not have GUI
2. Stand-alone machine
3. Lack of distributed system

BIBLIOGRAPHY:

1. Computer Science Class 12th Book
2. Class Notes