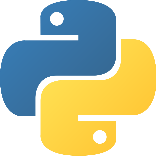
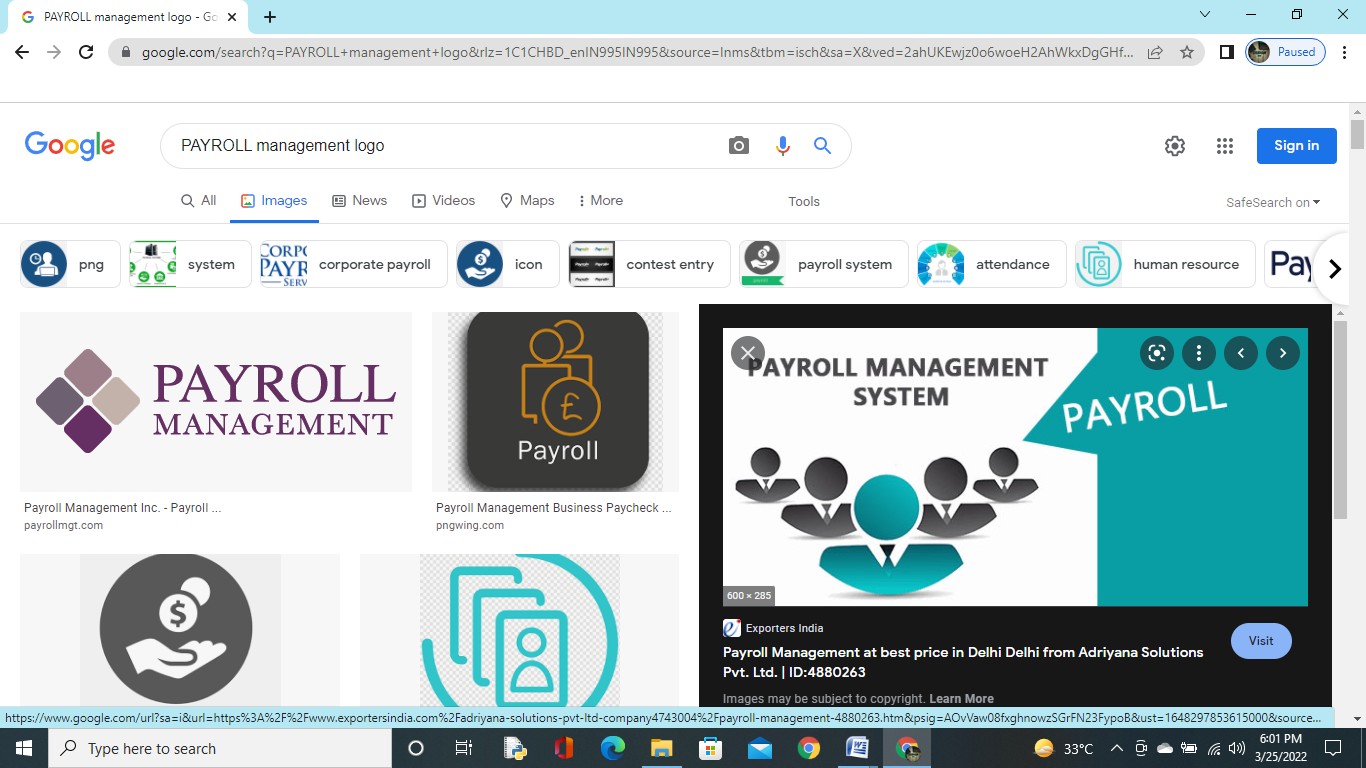
**PAYROLL MANAGEMENT SYSTEM**

***A Python Project***



* + **Employee Management**
  + **Saves Money and Time**
  + **Single Click Salary Processing**

**Python-SQL Connectivity**

**Database**

**Management Program**

CERTIFICATE

This is to certify that Shubham Yadav student of class 12th A has successfully completed their computer science python sql project on payroll management SYSTEM.

Mr. Mohd Aslam

(PGT Computer Science)

ACKNOWLEDGEMENT

I would like to express my Special thanks of Gratitude to my computer Science Teacher Mr. Mohd. Aslam for their able guidance and support in completing my project

I would like to extend my gratitude to my Parents and My Classmates who help me a lot in completing my project.

INDEX

|  |  |  |  |
| --- | --- | --- | --- |
| s**.NO**. | TITLE | PAGE NO. | SIGN. |
| 1. | CERTIFICATE | 3 |  |
| 2. | ACKNOWLEDGEMENT | 4 |  |
| 3. | INTRODUCTION | 6 |  |
| 4. | PROPOSED SYSTEM | 7 |  |
| 5. | FEATURES | 8 |  |
| 6. | APPLICATION | 9 |  |
| 7. | DESIGN | 10 |  |
| 8. | IMPLEMENTATION | 11-20 |  |
| 9. | OUTPUT | 21-24 |  |
| 10. | FUTURE IMPROVEMENTS | 25 |  |
| 14. | MY CONTRIBUTION | 26 |  |
| 15. | BIBLIOGRAPHY | 27 |  |
| 16. | ANNEXURE | 28 |  |
|  |  |  |  |

INTRODUCTION

A payroll management system is a software that

is used to manage employee’s financial record

in simple and automated way. It is solely based

on Python Programming Language along with

SQL connectivity.

The systems is the hassles of managing vast

amount of data by making the faculty

management and payroll processing easier.

The program manages your payroll in a

digitalized way according to user’s

consideration making it an easy to use

application.

PROPOSED SYSTEM

The application is designed in such a way to give user a better experience of using the app.It has well maintained table with statistics of SQL functionality to keep record of their payroll management.

Application is supposed to add, show and manage user all payroll database and tables to users choice data of his/her all daily basis requirements.

Moreover, application is designed in such a way to save your time by getting output directly in statistical way with simple input.

FEATURES

This program source many useful application that makes it easy to use.

Some most important applications can be defined as followed:

* + Database creation: user can create databases of employee.
  + Table creation: user can create tables of database.
  + Adding records: user can add records of employee in the table.
  + Salary slip maintain: user can maintain their employees salary slip according to date by date.
  + Calculation: user can add salary information of employee and can get net salary with calculation.

APPLICATIONS

This program can be used in various fields:

* + Employee salary processing.
  + Employee salary calculation and tax deduction.
  + Maintain pay slips and pay sheets.
  + Compute the salary to be paid for all employees.
  + Employee monthly salary report.
  + Maintain database and records of employee.

DESIGN

Program was made with python IDLE and SQL connectivity with basic knowledge.

* + Firstly the required modules were written on paper.
  + Then required output of program was drawn on paper.
  + Then logic was created how it should store information using SQL basic commands.
  + Code examples were studied.
  + At last, implementation was done.

IMPLEMENTATION

import mysql . connector

import datetime

from tabulate import tabulate

db=input("Enter name of your database : ")

mydb=mysql.connector.connect(host=' localhost' , user='root' , passwd='12345' )

mycursor = mydb.cursor()

sql="CREATE DATABASE if not exists %s" % (db,)

mycursor.execute (sql)

print("Database created Successfully..")

mycursor=mydb.cursor ( )

mycursor.execute ("Use "+db)

TableName=input ("Name of Table to be created:")

query="Create table if not exists "+TableName+" \

(empno int primary key,\

name varchar (15) not null,\

job varchar (15) ,\

BasicSalary int,\

DA float,\

HRA float,\

GrossSalary float,\

Tax float,\

NetSalary float)"

print ("Table "+TableName+" created successfully. ...")

mycursor.execute(query)

while True:

print('\n\n\n')

print("\*"\*95)

print('\t\t\t\t\tMAIN MENU')

print ("\*"\*95)

print("\t\t\t\t1. Adding Employee records")

print('\t\t\t\t2. For Displaying Record of All the Employee')

print('\t\t\t\t3. For Displaying Record of a particular Employee')

print('\t\t\t\t4. For Deleting Records of All the Employee')

print('\t\t\t\t5. For Deleting a Record of a particular Employees')

print('\t\t\t\t6. For Modification in a Record')

print('\t\t\t\t7. For Displaying Payroll')

print('\t\t\t\t8. For Displaying Salary Slip of All the Employees')

print('\t\t\t\t9. For Displaying Salary Slip of a particular Employee')

print('\t\t\t\t10.For Exit')

print('Enter Choice...',end='')

choice=int(input())

if choice==1:

try:

print('Enter employee information.....')

mempno=int (input(' Enter employee no:'))

mname=input(' Enter employee name:')

mjob=input ('Enter employee job:')

mbasic=float (input('Enter basic salary:'))

if mjob.upper()=='OFFICER':

mda=mbasic\*0.05

mhra=mbasic\*0.35

mtax=mbasic\*0.02

elif mjob.upper ( )=='MANAGER':

mda=mbasic\*0.45

mhra=mbasic\*0.30

mtax=mbasic\*0.15

else:

mda=mbasic\*0.40

mhra=mbasic\*0.25

mtax=mbasic\*0.1

mgross=mbasic+mda+mhra

mnet=mgross-mtax

rec=(mempno, mname, mjob, mbasic, mda, mhra, mgross, mtax, mnet)

query="insert into "+TableName+" values(%s,%s,%s,%s,%s,%s,%s,%s,%s)"

mycursor.execute (query, rec)

mydb.commit()

print ('Record added successfully....')

except Exception as e:

print('Something went wrong',e)

elif choice==2:

try:

query='select\*from '+TableName

mycursor.execute(query)

print(tabulate(mycursor, headers=['EmpNO','Name','job','Basic Salary','DA','HRA','GROSS Salary','Tax','Net Salary'],tablefmt='fancy\_grid'))

'''myrecords=mycursor.fetchall()

for rec in myrecords:

print(rec)'''

except:

print('something went wrong')

elif choice==3:

try:

en=(input('Enter Employee no.of the record to be displayed...'))

query='select\*from '+TableName+' Where empno='+en

mycursor.execute(query)

print("\n\nRecord of Employee no.: "+en)

print(tabulate(mycursor, headers=['EmpNO','Name','job','Basic Salary','DA','HRA','GROSS Salary','Tax','Net Salary'],tablefmt='fancy\_grid'))

c=mycursor.rowcount

if c==0:

print('Nothing to display')

except:

print('something went wrong')

elif choice==4:

try:

ch=input('Do you want to delete all the records (YES/NO)')

if ch.upper()=='YES':

mycursor.execute('delete from '+TableName)

mydb.commit()

print('All the records are deleted...')

except:

print('semething went wrong')

elif choice==5:

try:

en=input('Enter employee no. of the record to be deleted... ')

query='delete from '+TableName+' where empno='+en

mycursor.execute(query)

mydb.commit()

c=mycursor.rowcount

if c>0:

print('Deletion done')

else:

print('Employee no.',en,'not found')

except:

print("something went wrong")

elif choice==6:

try:

en=input('Enter employee no. of the record to be modified... ')

query='select\*from '+TableName+' where empno='+en

mycursor.execute(query)

myrecord=mycursor.fetchone()

c=mycursor.rowcount

if c==0:

print('Empno '+en+' does not exist')

else:

mname=myrecord[1]

mjob=myrecord[2]

mbasic=myrecord[3]

print('empno :',myrecord[0])

print('name :',myrecord[1])

print('job :',myrecord[2])

print('basic :',myrecord[3])

print('DA :',myrecord[4])

print('HRA :',myrecord[5])

print('Gross :',myrecord[6])

print('Tax :',myrecord[7])

print('Net :',myrecord[8])

print('\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_')

print('Type Value to modify below or just press Enter for no change')

x=input('Enter name ')

if len(x)>0:

mname=x

x=input('Enter job ')

if len(x)>0:

mjob=x

x=input('Enter basic salary ')

if len(x)>0:

mbasic=float(x)

query='update '+TableName+' set name='+"'"+mname+"'"+','+'job='+"'"+mjob+"'"+','+'basicsalary='+str(mbasic)+' where empno='+en

print(query)

mycursor.execute(query)

mydb.commit()

print('Record Modified')

except:

print('Something went wrong')

elif choice==7:

try:

query='select\*from '+TableName

mycursor.execute(query)

myrecords=mycursor.fetchall()

print("\n\n\n")

print(95\*'\*')

print('Employee Payroll'.center(90))

print(95\*'\*')

now=datetime.datetime.now()

print("Current Date and Time:",end=' ')

print(now.strftime("%Y-%m-%d %H:%M:%S"))

print()

print(95\*'-')

print('%-5s %-15s %-10s %-8s %-8s %-10s %-9s %-8s %-9s'\

%('Empno','Name','Job','Basic','DA','HRA','Gross','Tax','Net'))

print(95\*'-')

for rec in myrecords:

print('%4d %-15s %-10s %8.2f %8.2f %9.2f %9.2f %8.2f %9.2f'%rec)

print(95\*'-')

except:

print('Something went wrong')

elif choice==8:

try:

query='select \* from '+TableName

mycursor.execute(query)

now=datetime.datetime.now()

print("\n\n\n")

print(95\*'-')

print("\t\t\t\tSalary Slip")

print(95\*'-')

print("Current Date and Time:",end=' ')

print(now.strftime("%Y-%m-%d %H:%M:%S"))

myrecords=mycursor.fetchall()

for rec in myrecords:

print('%4d %-15s %-10s %8.2f %8.2f %9.2f %9.2f %8.2f %9.2f'%rec)

except:

print('Something went wrong')

elif choice==9:

try:

en=input('Enter employee number whose pay slip you want to retreieve:')

query='select \* from '+TableName+' where empno='+en

mycursor.execute(query)

now=datetime.datetime.now()

print("\n\n\n\t\t\t\tSalary Slip ")

print("Current Date and Time:",end=' ')

print(now.strftime("%Y-%m-%d %H:%M:%S"))

print(tabulate(mycursor,headers=['Empno','Name','Job','Basic Salary','DA','HRA','Gross salary','Tax','Net Salary'],tablefmt='fancy\_grid'))

except Exception as e:

print('something went wrong',e)

elif choice==10:

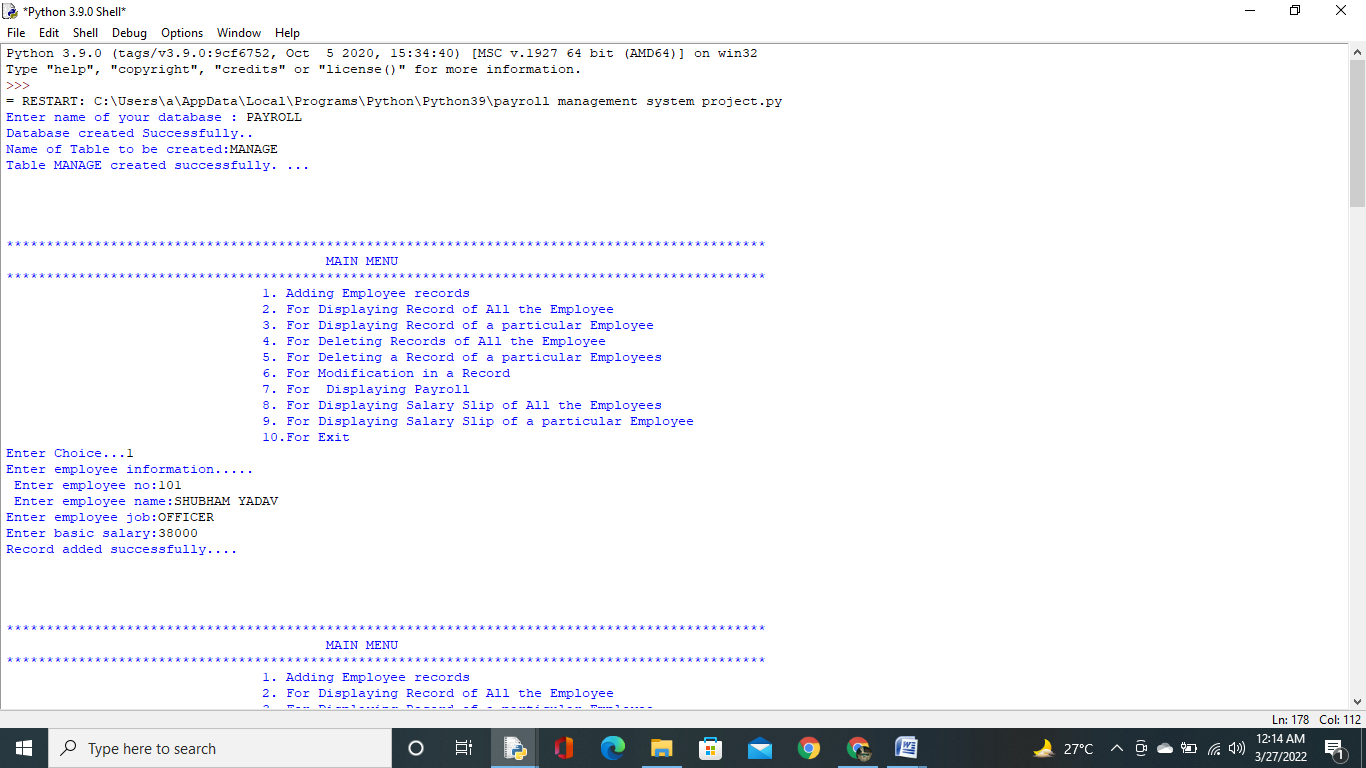
break

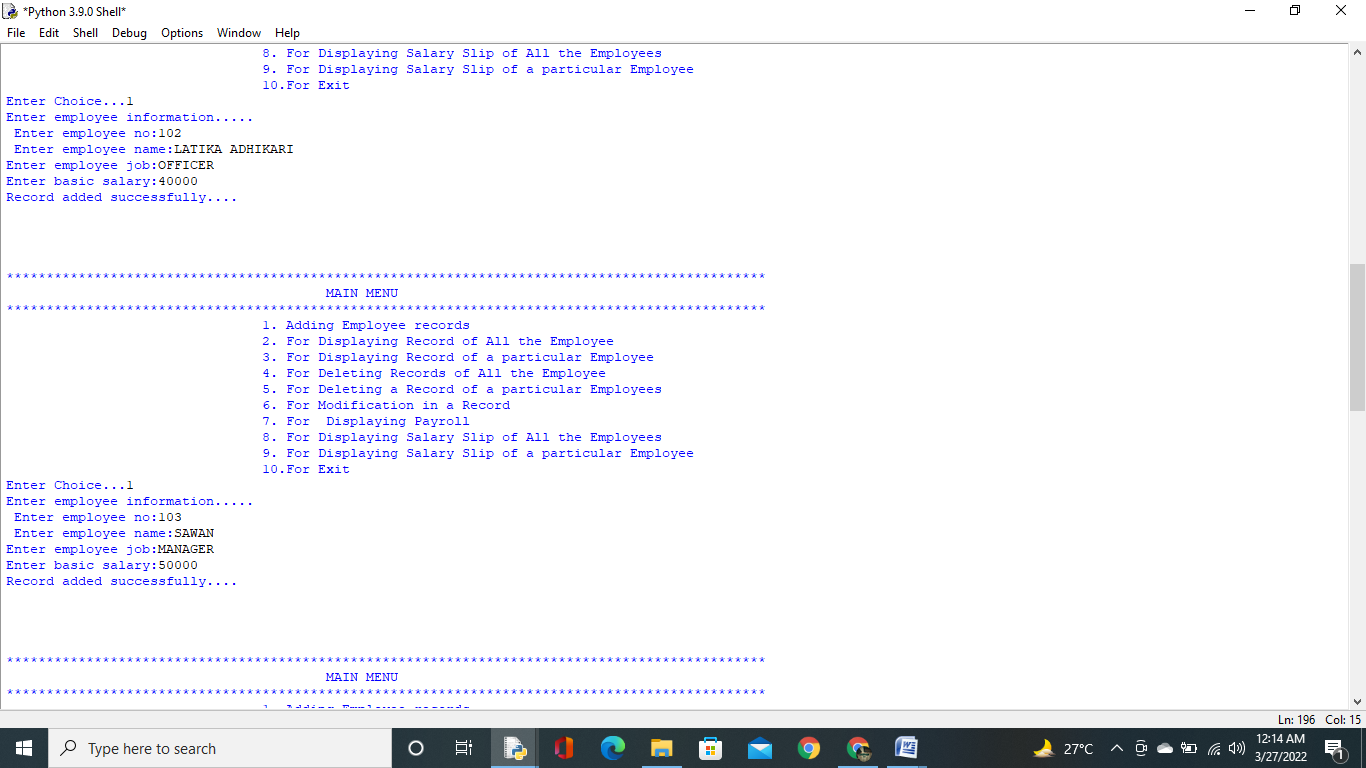
else:

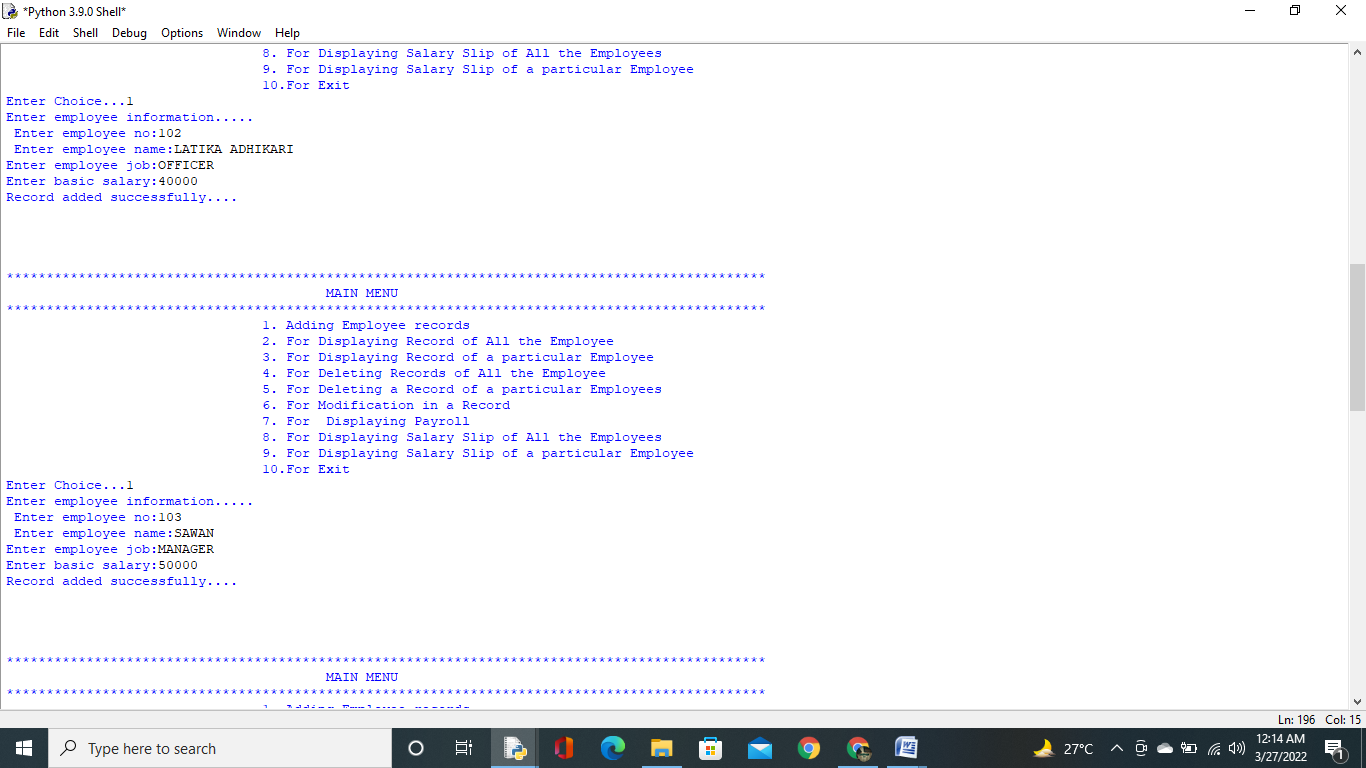
print('wrong choice.....')

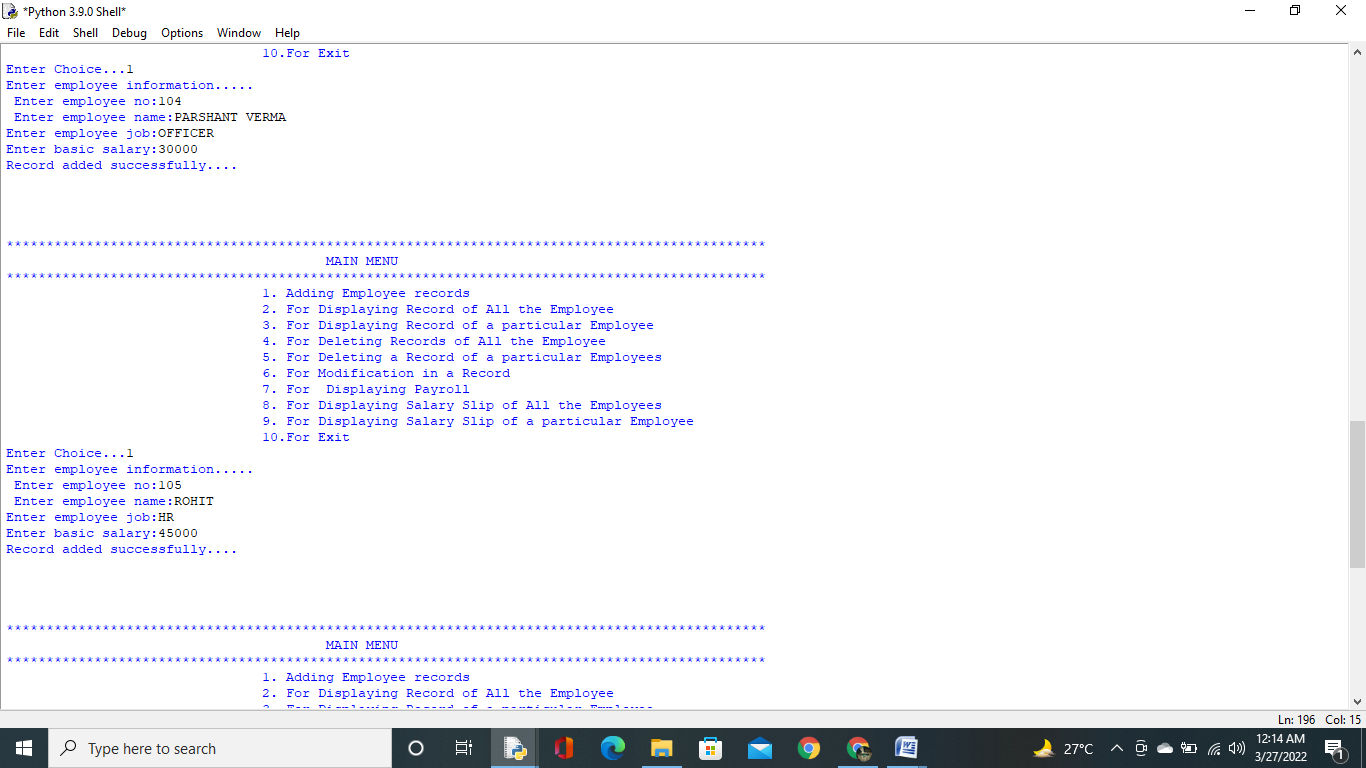
OUTPUTS

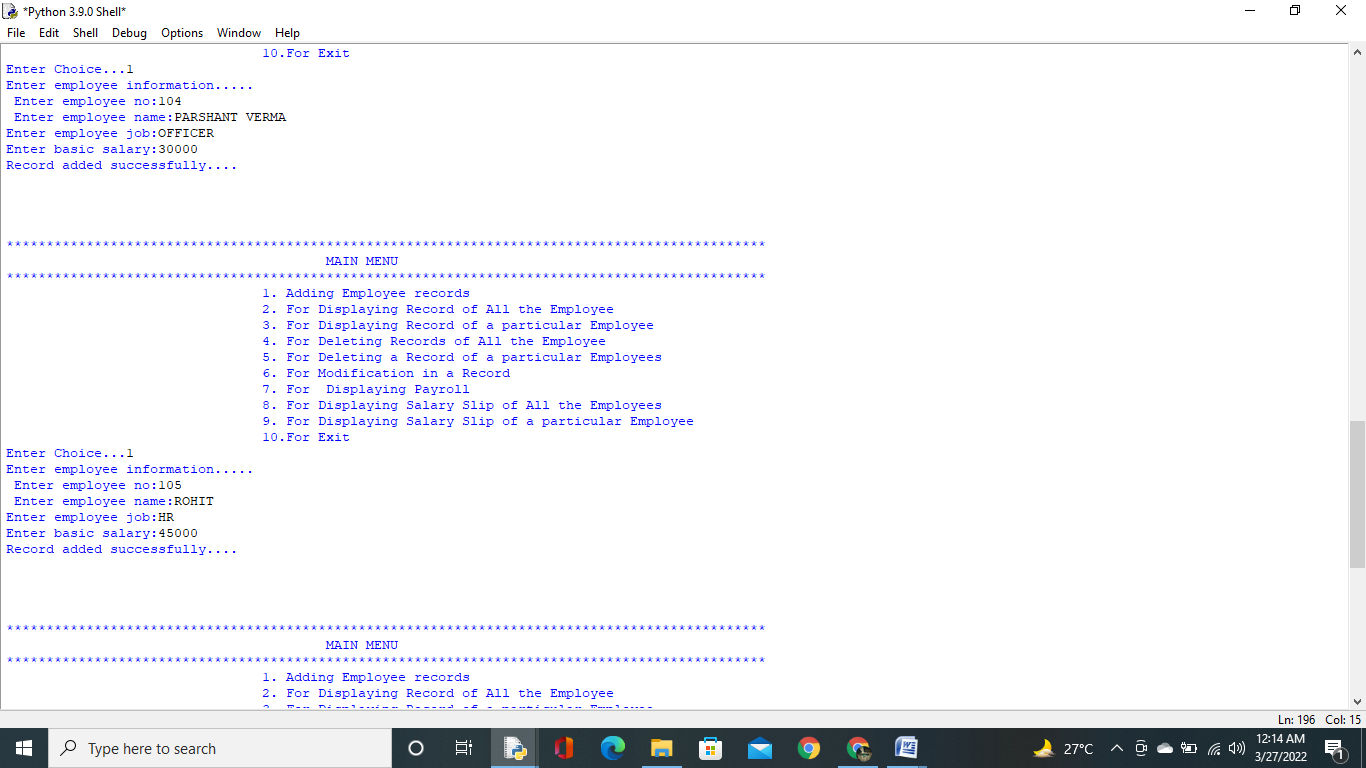
OUTPUT FOR PRESS 1.



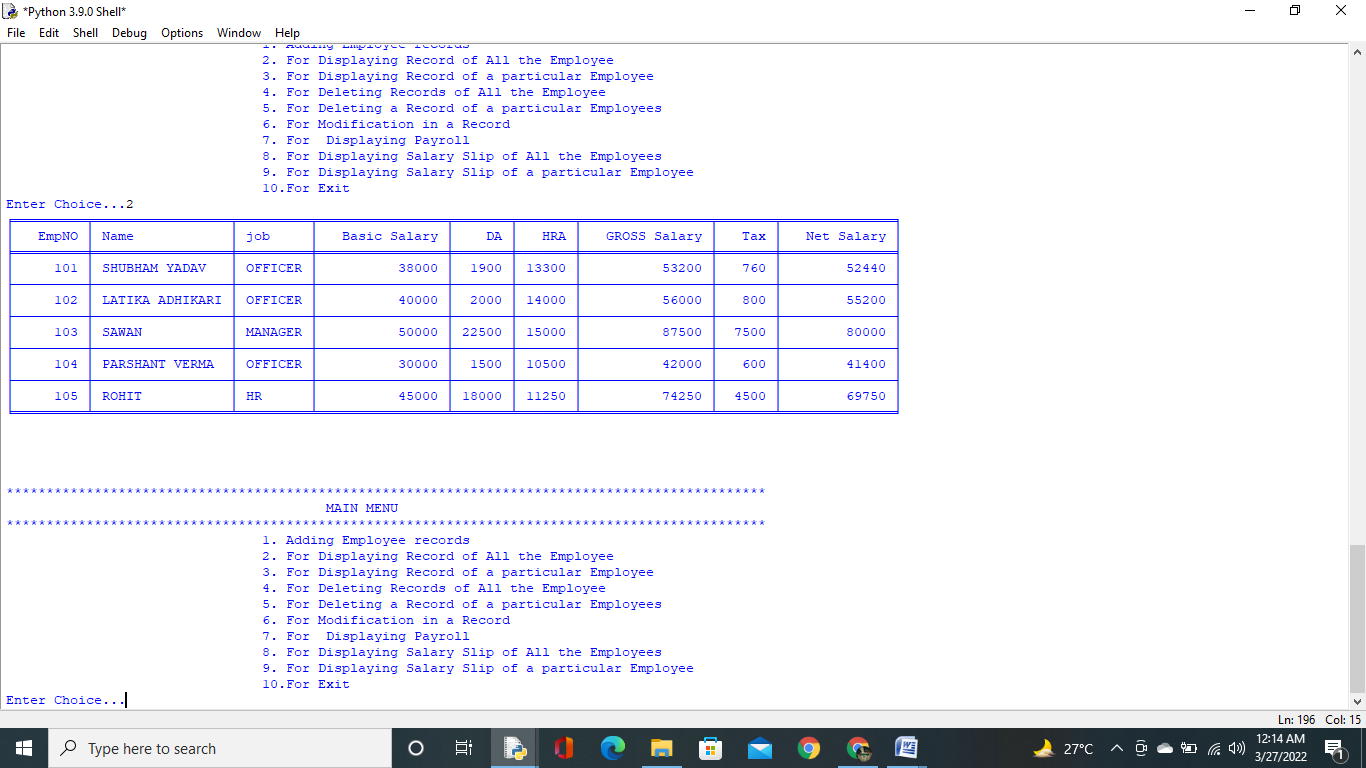




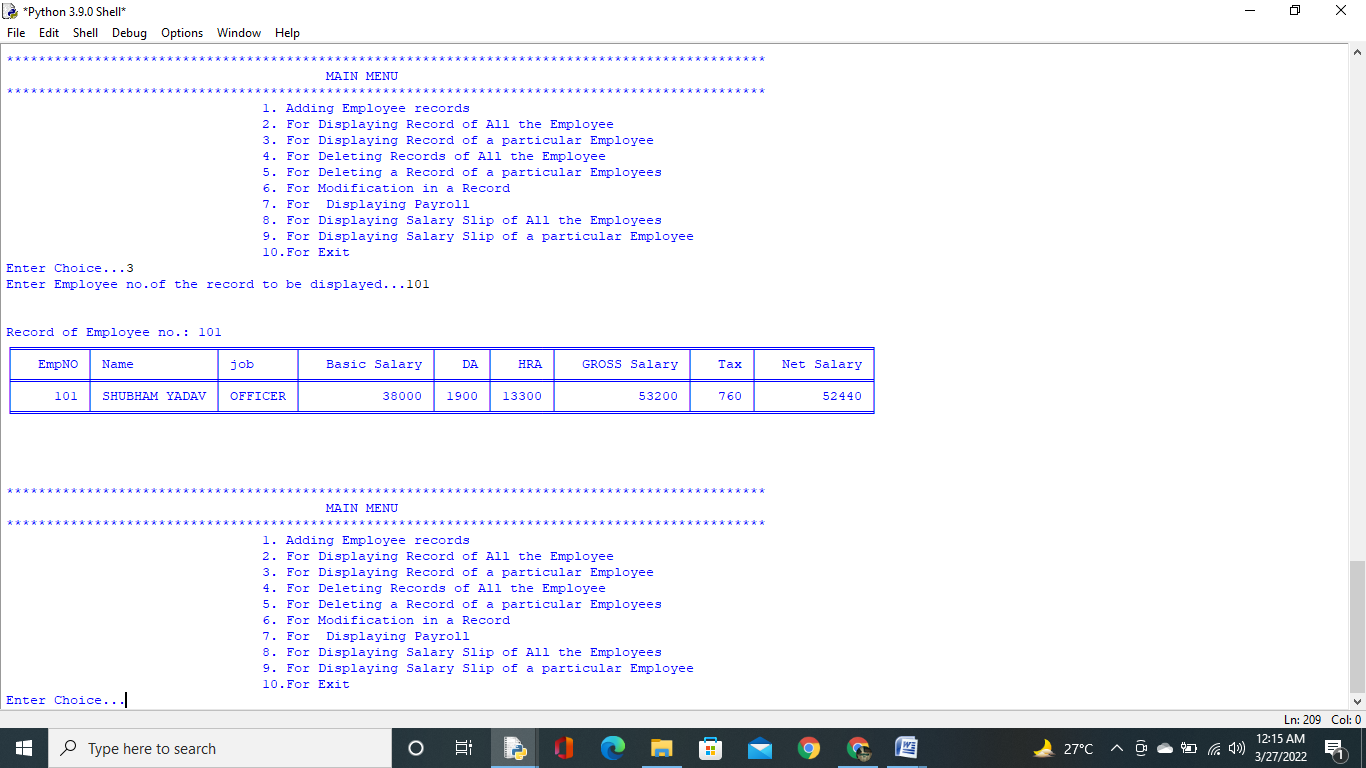




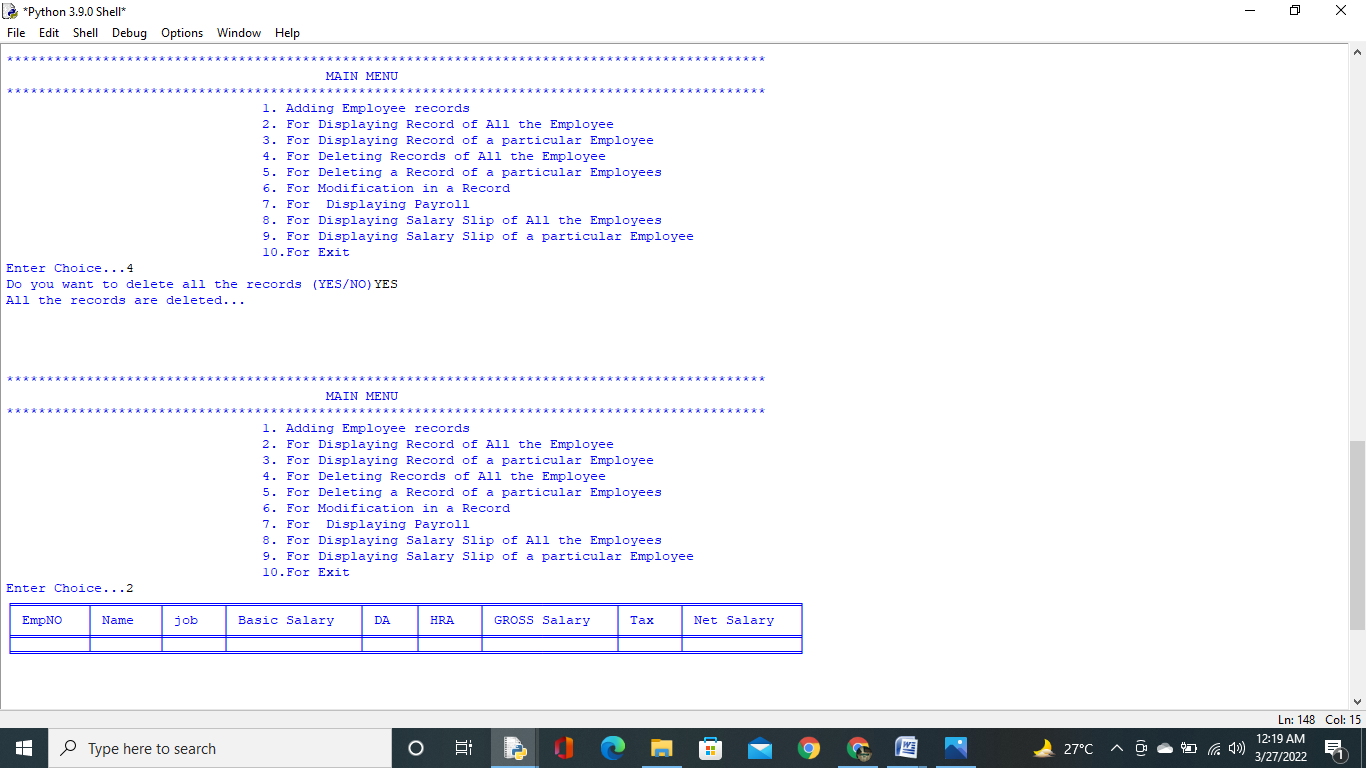
OUTPUT FOR PRESS 2.



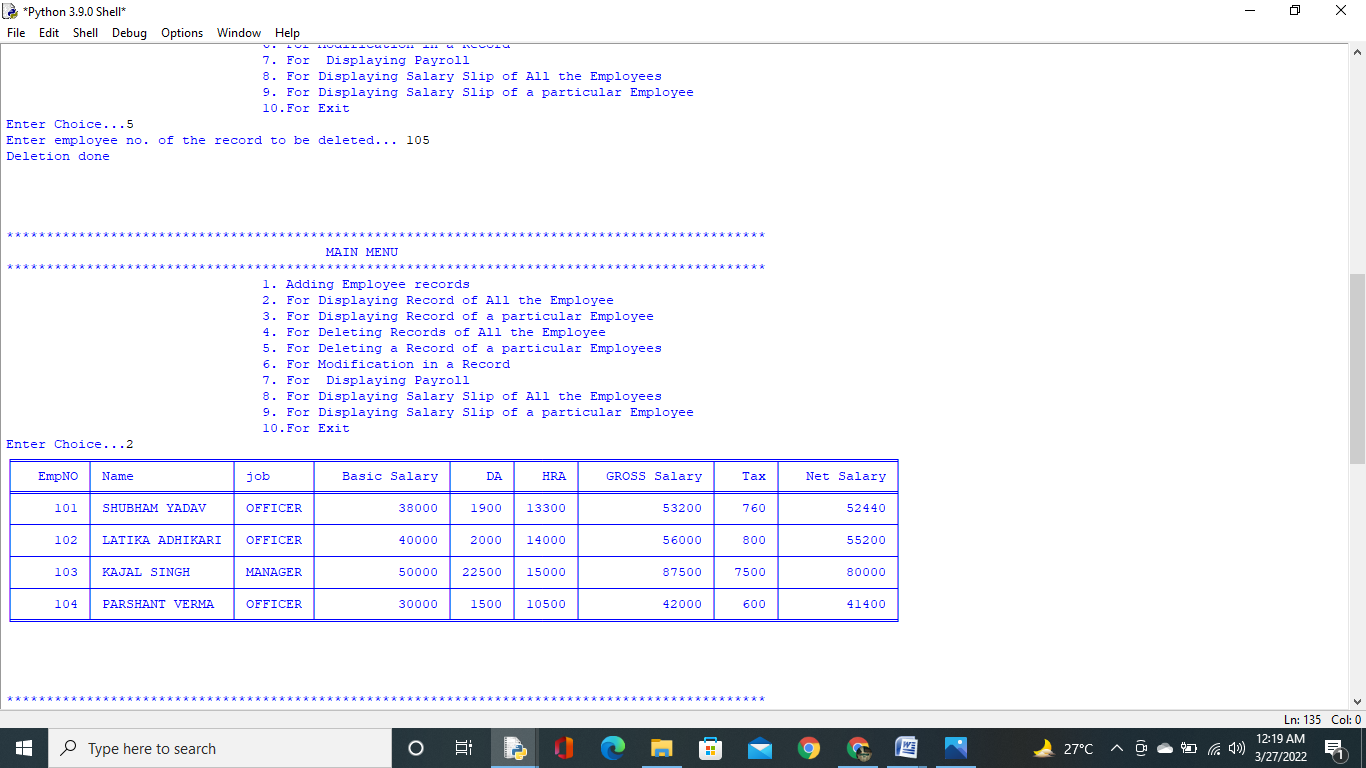
OUTPUT FOR PRESS 3.



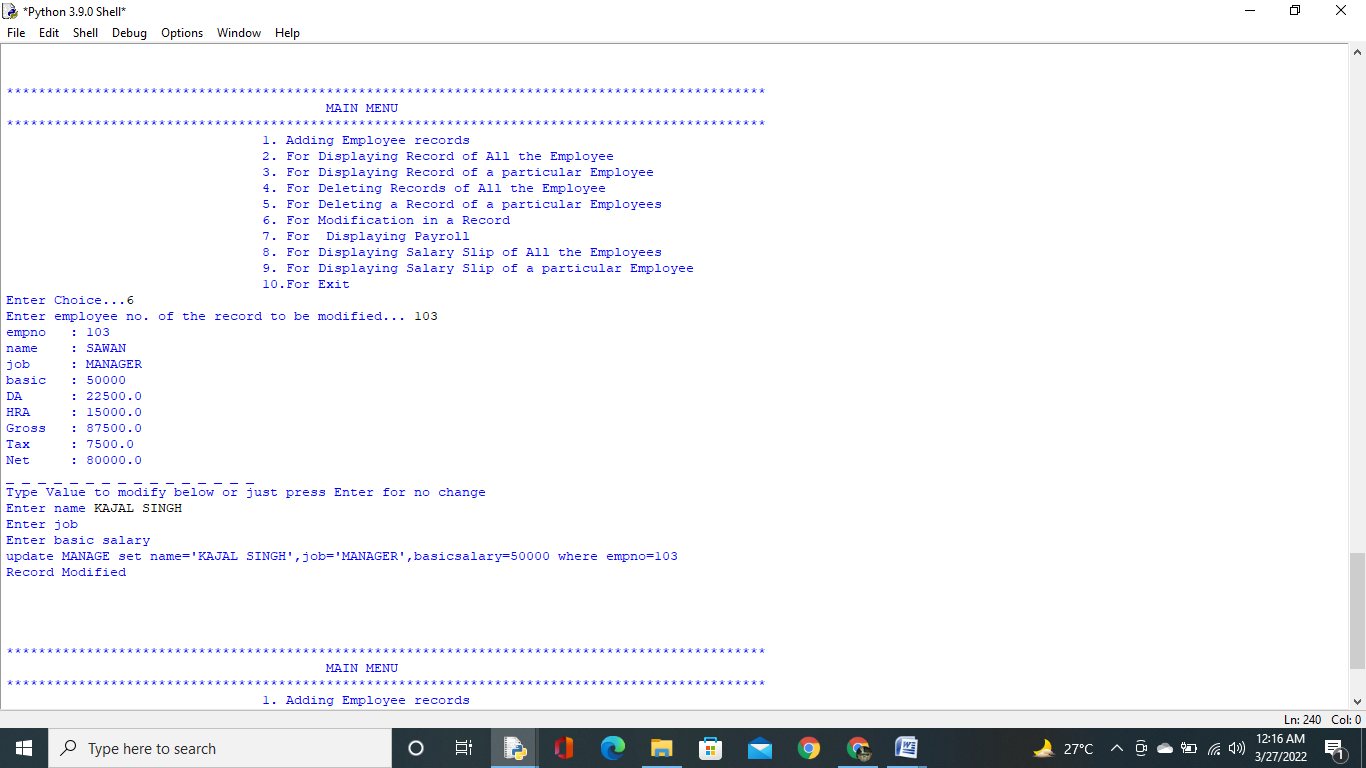
OUTPUT FOR PRESS 4.

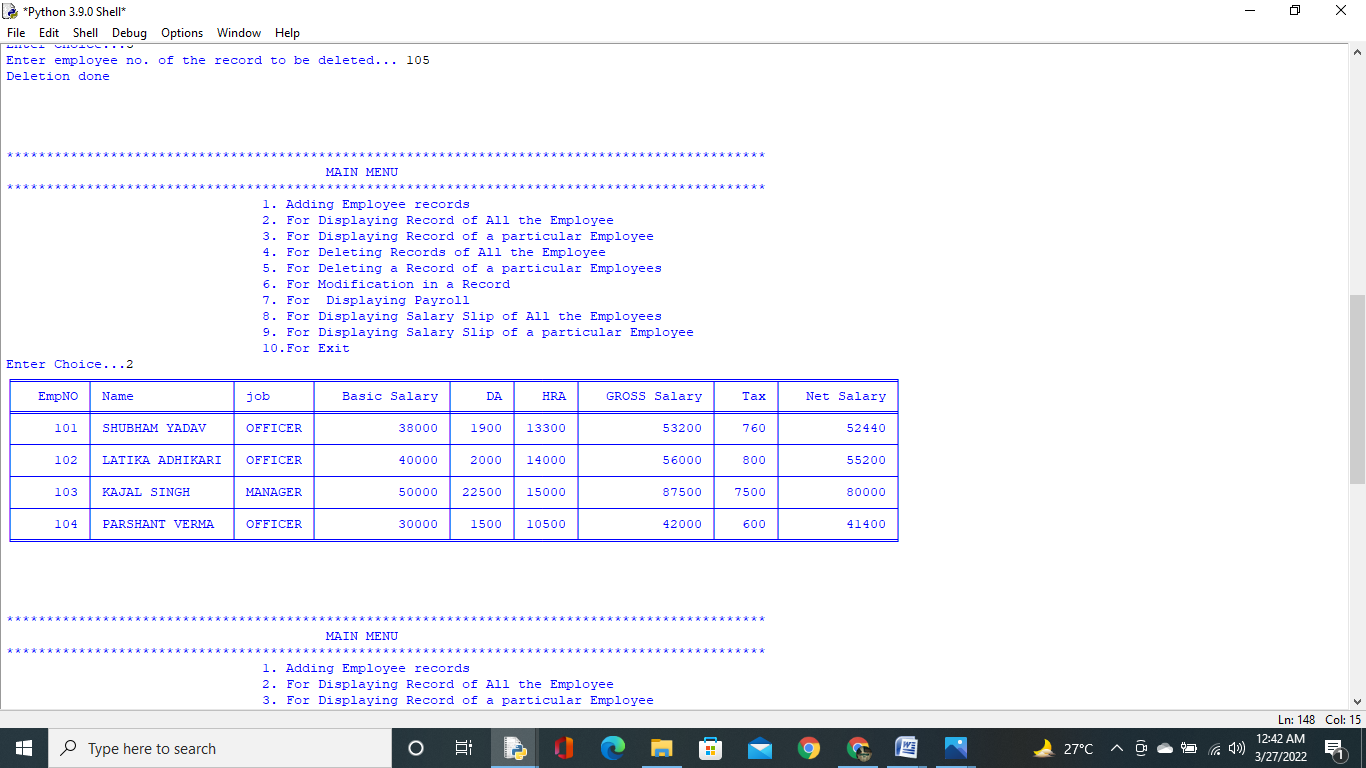


OUTPUT FOR PRESS 5.

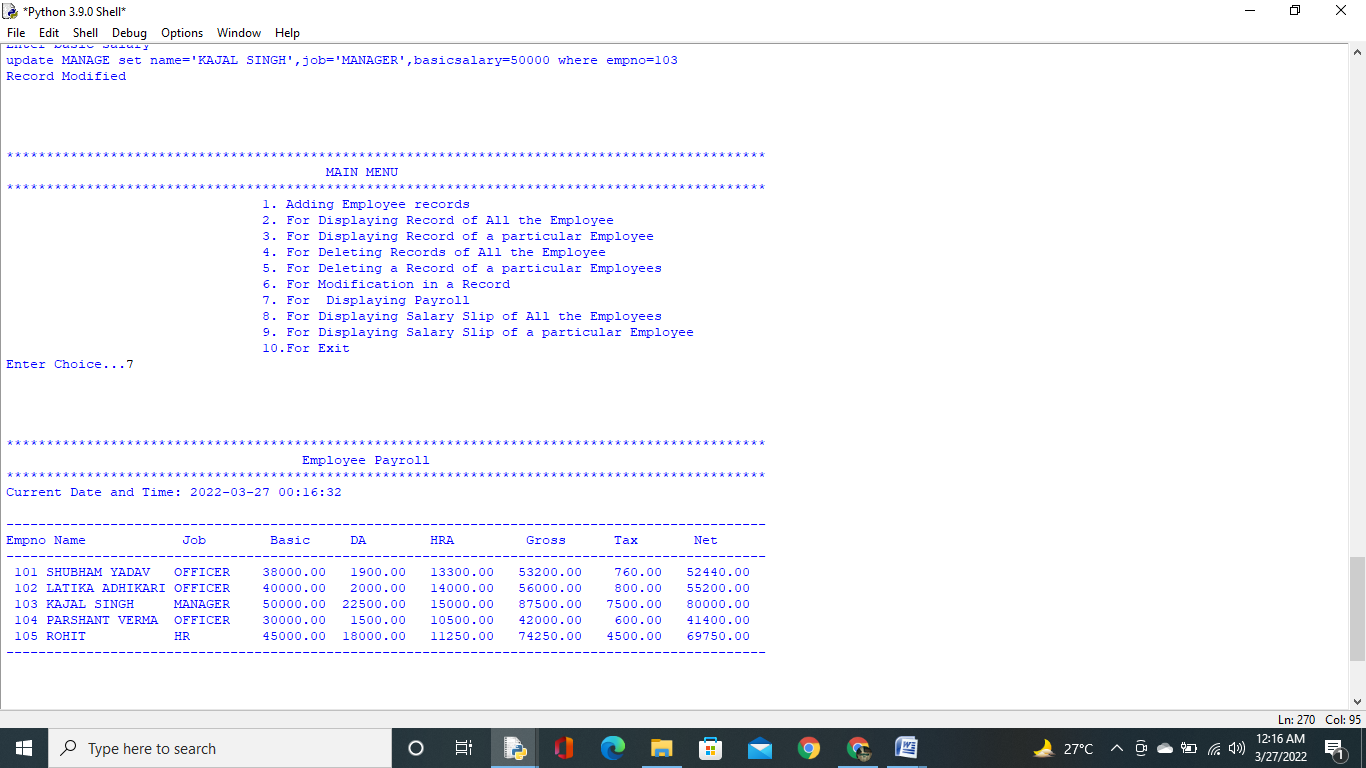


OUTPUT FOR PRESS 6.

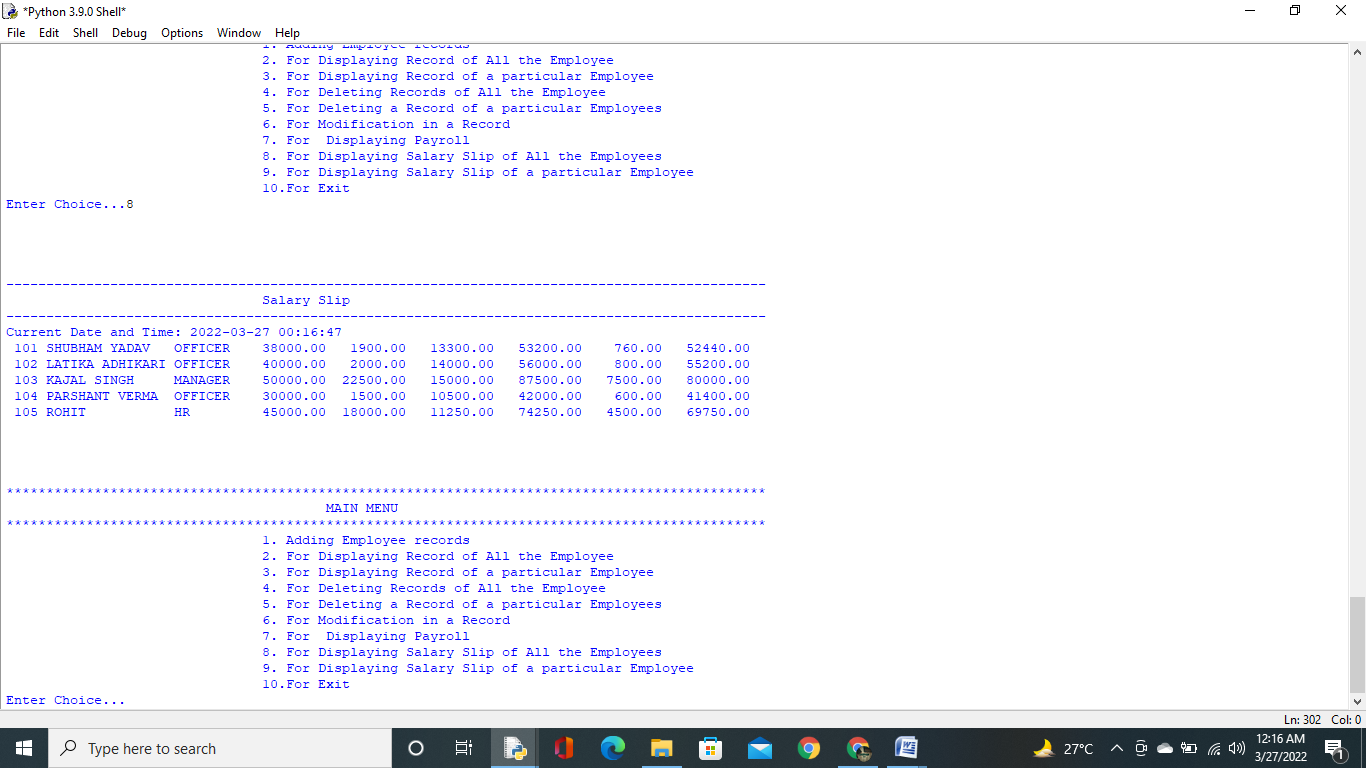




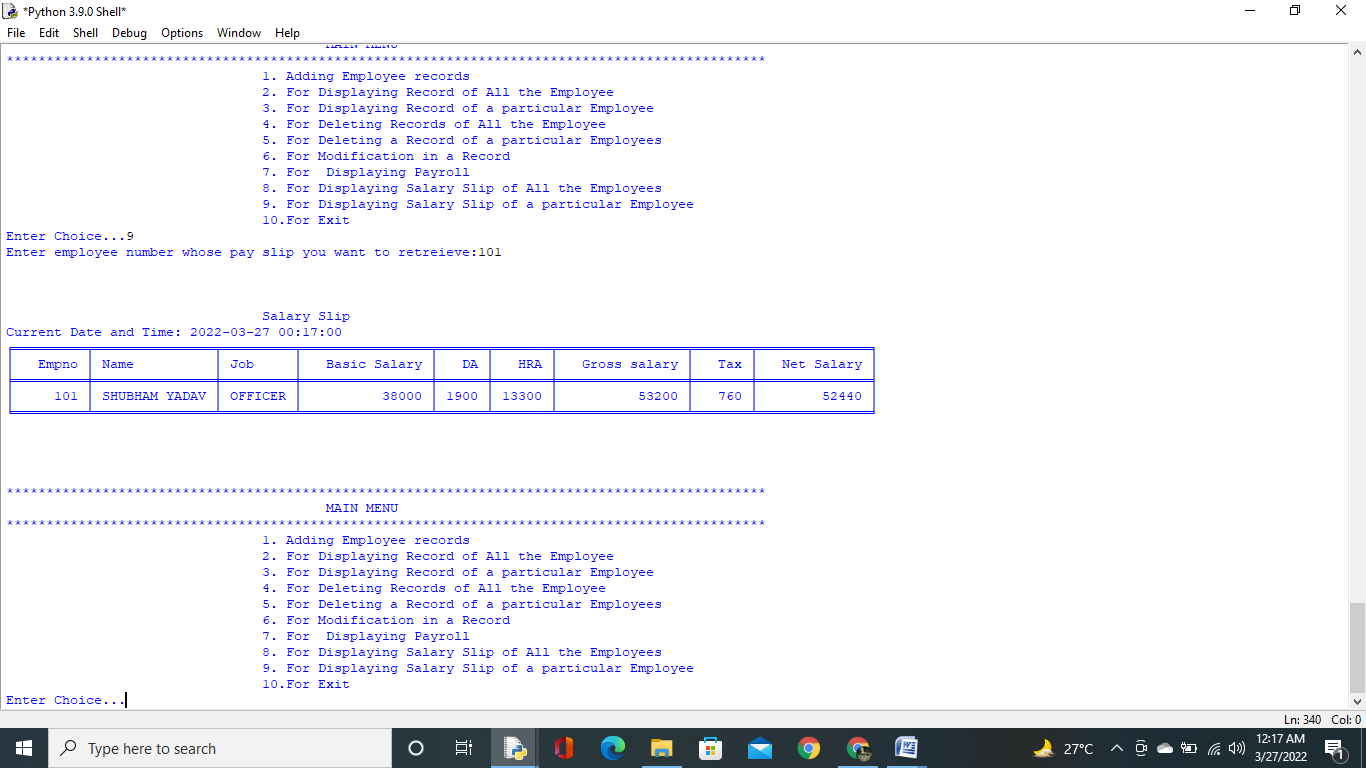
OUTPUT FOR PRESS 7.



OUTPUT FOR PRESS 8.



OUTPUT FOR PRESS 9.



OUTPUT FOR PRESS 10.

EXIT FROM PROGRAM.

FUTURE IMPROVEMENTS

* Adding GUI(GRAPHICS USER INTERFACE).
* Adding login user interface.
* Adding hints for first time users.
* Adding security feature to data.
* Adding attendance report.

MY CONTRIBUTION

SHUBHAM YADAV:

My task was to look for the error in the program and correct it. Tutorials from YouTube helped me to learn

about it. Selecting logic for outputs such as resultant

data in tabular form and data for a particular employee.

I also helped in making of project file with my partner.

BIBLIOGRAPHY

1. *Arora,Preeti. Computer Science with Python Class XII . Delhi, Sultan Chand publication, 2021*

1. *Arora,Sumita. Computer Science with Python Class XII . Delhi, Dhanpat Rai & Co. publication, 2021*

1. Interface Python with SQL tutorial

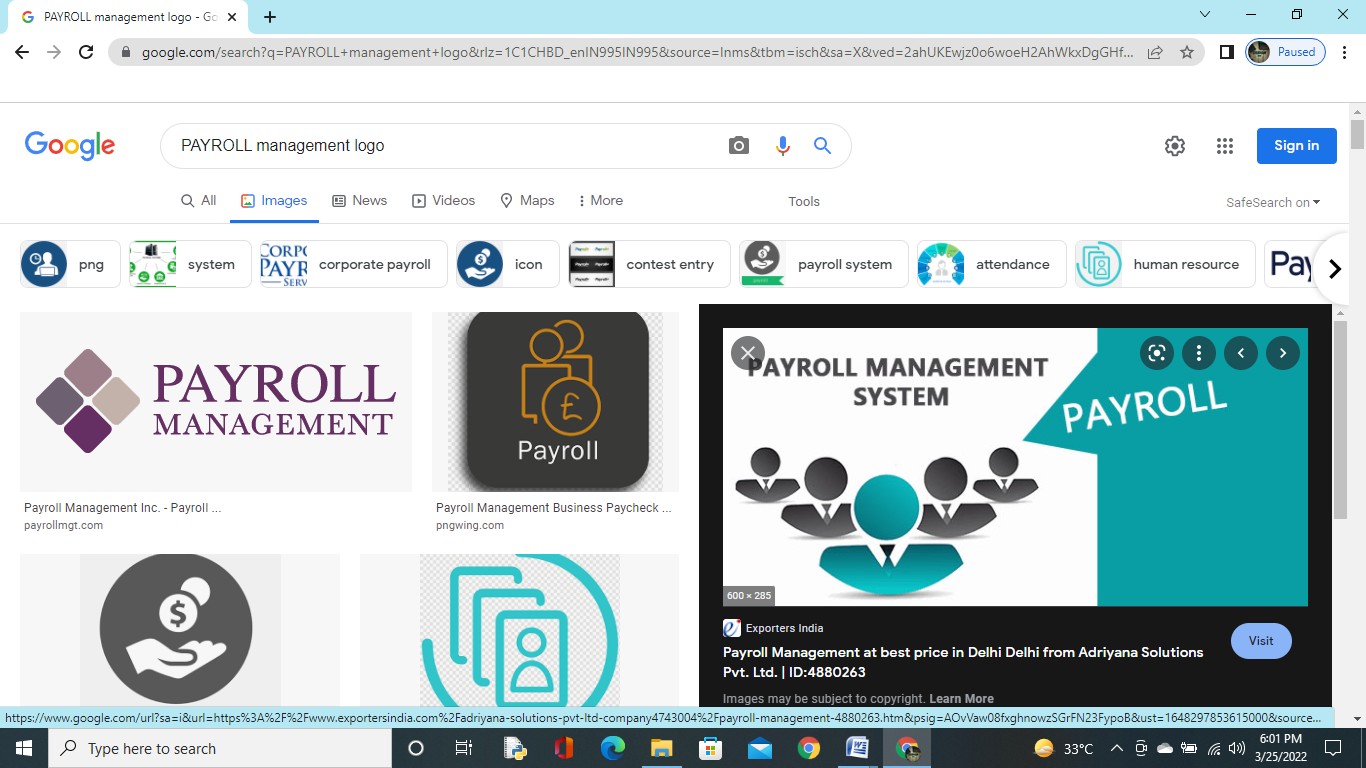
<https://www.youtube.com/watch?v=MnVsxQd2bKg>

1. Tabulate programming tutorial

<https://www.youtube.com/watch?v=QjIqYbiVQuw>

ANNEXURE

Images used in the program are as follows:



Logo image

<https://insightinfosystem.com/Billing_ERP_software>