Aakshat Malhotra || 2021UCS1557

Problem Statement:

To make a salary/payroll management System

Requirement Analysis

1. In this system we can add details about employees
2. We can update paybill on monthly basis.
3. We can remove employee and their payroll data
4. This system automatically calculates various allowances and deductions in the salary.

Tables Used

1. 3 tables are used
2. Employee table to store employee information

   Structure of Employee Table:

```
mysql> desc emp;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| ECODE  | int         | NO   | PRI | NULL    |       |
| FNAME  | varchar(20) | NO   |     | NULL    |       |
| LNAME  | varchar(20) | NO   |     | NULL    |       |
| DESIG  | char(15)    | NO   |     | NULL    |       |
| GENDER | char(1)     | YES  |     | M       |       |
| DOB    | date        | YES  |     | NULL    |       |
| DOJ    | date        | YES  |     | NULL    |       |
| MOB    | varchar(11) | YES  |     | NULL    |       |
| PAN    | char(10)    | YES  |     | NULL    |       |
| ACNO   | varchar(15) | YES  |     | NULL    |       |
| IFSC   | char(11)    | YES  |     | NULL    |       |
| BASIC  | int         | YES  |     | NULL    |       |
| TA     | int         | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
```

3. Pay table to store the paybill

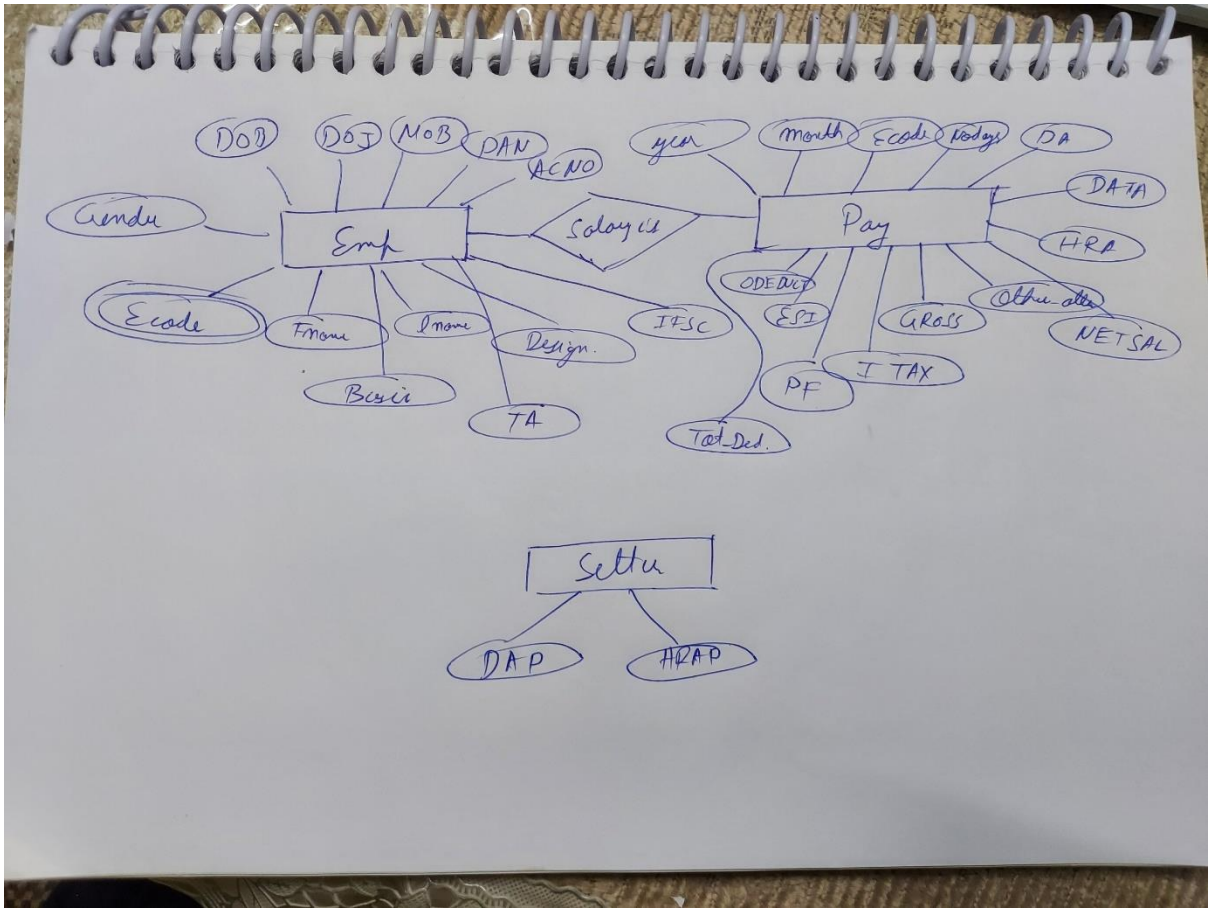
   Structure of Pay Table:

```
mysql> desc pay;
+------------+------+------+-----+---------+-------+
| Field      | Type | Null | Key | Default | Extra |
+------------+------+------+-----+---------+-------+
| YEAR       | int  | NO   | PRI | NULL    |       |
| MONTH      | int  | NO   | PRI | NULL    |       |
| ECODE      | int  | NO   | PRI | NULL    |       |
| NODAYS     | int  | NO   |     | NULL    |       |
| DA         | int  | YES  |     | NULL    |       |
| DATA       | int  | YES  |     | NULL    |       |
| HRA        | int  | YES  |     | NULL    |       |
| OTHER_ALLW | int  | YES  |     | NULL    |       |
| GROSS      | int  | YES  |     | NULL    |       |
| ITAX       | int  | YES  |     | NULL    |       |
| PF         | int  | YES  |     | NULL    |       |
| ESI        | int  | YES  |     | NULL    |       |
| ODEDUCT    | int  | YES  |     | NULL    |       |
| TOT_DEDUC  | int  | YES  |     | NULL    |       |
| NETSAL     | int  | YES  |     | NULL    |       |
+------------+------+------+-----+---------+-------+
15 rows in set (0.00 sec)
```

4. Setter to store the current DA and Hra percentage
   Structure of Setter Table:

```
mysql> desc setter;
+-------+------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+-------+------+------+-----+---------+-------+
| DAP   | int  | YES  |     | NULL    |       |
| HRAP  | int  | YES  |     | NULL    |       |
+-------+------+------+-----+---------+-------+
2 rows in set (0.03 sec)

mysql>
```

ER Diagram



Create Table using queries in python

```
def __init__(self):
    self.con = connector.connect(host="localhost", user="root", passwd="password", database="aakshat")

    query1 = "create table if not exists EMP(ECODE INT(6) PRIMARY KEY, FNAME VARCHAR(20) NOT NULL, LNAME VARCHAR(20) NOT NULL,DESIG CHAR(15) NOT
    NULL, GENDER CHAR DEFAULT 'M', DOB DATE, DOJ DATE, MOB VARCHAR(11), PAN CHAR(10), ACNO VARCHAR(15), IFSC CHAR(11),BASIC INT(6), TA INT(4));"

    query2 = "create table if not exists PAY(YEAR INT(4), MONTH INT(2), ECODE INT(6), NODAYS INT(2) NOT NULL, DA INT(6), DATA INT(5), HRA INT(5),
    OTHER_ALLW INT(5), GROSS INT(6), ITAX INT(6), PF INT(6), ESI INT(6), ODEDUCT INT(5), TOT_DEDUC INT(7), NETSAL INT(7), PRIMARY KEY(YEAR, MONTH,
    ECODE), FOREIGN KEY (ECODE) REFERENCES EMP(ECODE));"

    query3 = "create table if not exists SETTER(DAP INT(3), HRAP INT(2));"



    cur = self.con.cursor();

    cur.execute(query1);
    cur.execute(query2);
    cur.execute(query3);
    self.con.commit();
    cur.execute("delete from setter");
    self.con.commit();
    cur.execute("insert into setter values(9, 8);")
    self.con.commit();
```

Insert values queries in python

Values are used inserted by console

Python Code

```
import mysql.connector as connector

class DBHelper:
    def __init__(self):
        self.con = connector.connect(host="localhost", user="root", passwd="password",
database="aakshat")

        query1 = "create table if not exists EMP(ECODE INT(6) PRIMARY KEY, FNAME
VARCHAR(20) NOT NULL, LNAME VARCHAR(20) NOT NULL,DESIG CHAR(15) NOT NULL, GENDER CHAR
DEFAULT 'M', DOB DATE, DOJ DATE, MOB VARCHAR(11), PAN CHAR(10), ACNO VARCHAR(15), IFSC
CHAR(11),BASIC INT(6), TA INT(4));"

        query2 = "create table if not exists PAY(YEAR INT(4), MONTH INT(2), ECODE
INT(6), NODAYS INT(2) NOT NULL, DA INT(6), DATA INT(5), HRA INT(5), OTHER_ALLW INT(5),
GROSS INT(6), ITAX INT(6), PF INT(6), ESI INT(6), ODEDUCT INT(5), TOT_DEDUC INT(7),
NETSAL INT(7), PRIMARY KEY(YEAR, MONTH, ECODE), FOREIGN KEY (ECODE) REFERENCES
EMP(ECODE));"

        query3 = "create table if not exists SETTER(DAP INT(3), HRAP INT(2));"



        cur = self.con.cursor();

        cur.execute(query1);
        cur.execute(query2);
        cur.execute(query3);
        self.con.commit();
        cur.execute("delete from setter");
        self.con.commit();
        cur.execute("insert into setter values(9, 8);")
        self.con.commit();


    def input_employee(self, ecode, fname, lname, desgination, gender, dob, doj, mob,
pan, acno, ifsc, basic, ta):
        cur = self.con.cursor();
        query = f"insert into emp values({ecode}, '{fname}', '{lname}',
'{desgination}', '{gender}', '{dob}','{doj}','{mob}','{pan}','{acno}','{ifsc}',
{basic}, {ta})";

        cur.execute(query);
        self.con.commit();
        print("Data entered successfully")

    def display_AllEmp(self):
        cur = self.con.cursor();
        query = "select * from emp;";

        cur.execute(query);

        for row in cur:
            print()
```

```
            print(f"ECODE: {row[0]}")
            print(f"First Name: {row[1]}")
            print(f"Last Name: {row[2]}")
            print(f"Designation: {row[3]}")
            print(f"Gender: {row[4]}")
            print(f"DOB: {row[5]}")
            print(f"DOJ: {row[6]}")
            print(f"MOB: {row[7]}")
            print(f"PAN: {row[8]}")
            print(f"ACNO: {row[9]}")
            print(f"IFSC: {row[10]}")
            print(f"Basic Pay: {row[11]}")
            print(f"TA: {row[12]}")
            print()
            print()

    def display_SpecificEmp(self, ecode):
        cur = self.con.cursor();
        query = f"select * from emp where ecode = {ecode}";

        cur.execute(query);

        for row in cur:
            print();
            print(f"ECODE: {row[0]}")
            print(f"First Name: {row[1]}")
            print(f"Last Name: {row[2]}")
            print(f"Designation: {row[3]}")
            print(f"Gender: {row[4]}")
            print(f"DOB: {row[5]}")
            print(f"DOJ: {row[6]}")
            print(f"MOB: {row[7]}")
            print(f"PAN: {row[8]}")
            print(f"ACNO: {row[9]}")
            print(f"IFSC: {row[10]}")
            print(f"Basic Pay: {row[11]}")
            print(f"TA: {row[12]}")
            print()

    def percentage_setter(self, daPercent, HRAPercent):
        cur = self.con.cursor();
        cur.execute("delete from setter")
        cur.execute(f"insert into setter values ({daPercent}, {HRAPercent})")
        self.con.commit();

    def show_rates(self):
        cur = self.con.cursor();
        cur.execute("select * from setter")
        dap= 0
        hrp = 0
        for row in cur:
            dap = row[0]
            hrp = row[1]
```

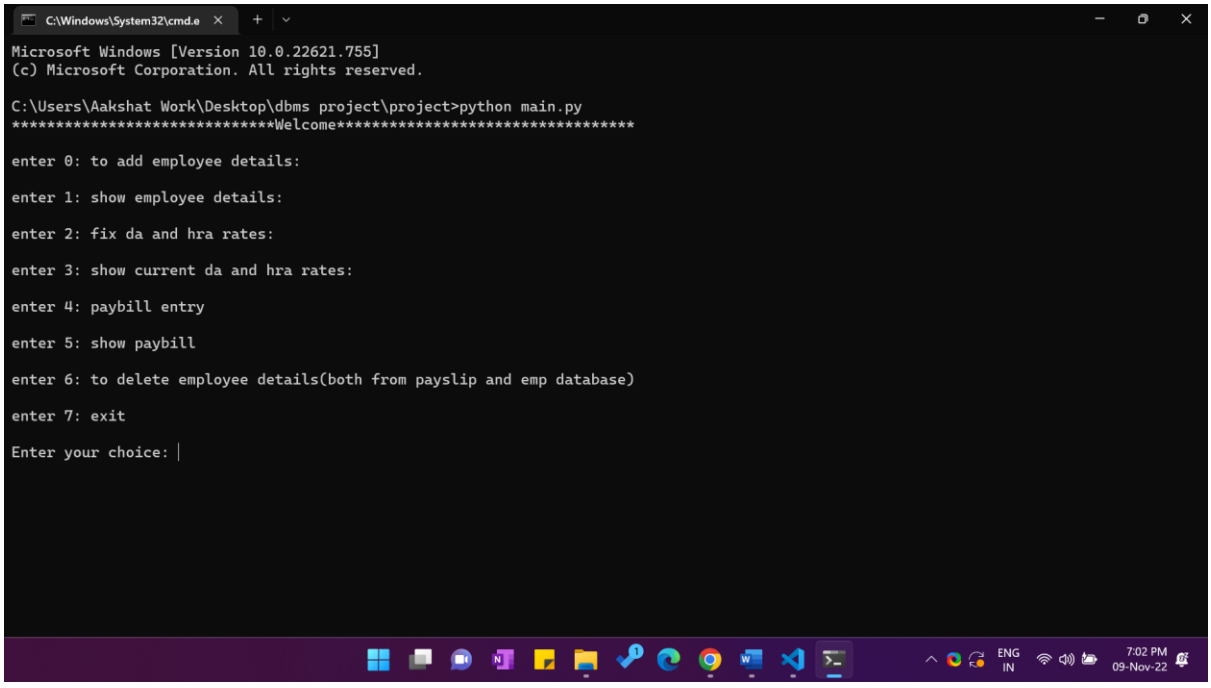
```
        print(f"DA percentage is {dap} \nHRA percentage is {hrp}")

    def salary_entryInd(self, year, month, ecode, no_of_days, da, DATA, hra,
other_allow, gross, itax, pf, esi, odeduct, tot_deduc, netsal):
        cur = self.con.cursor();
        cur.execute(f"insert into pay values({year},{month}, {ecode}, {no_of_days},
{da}, {DATA}, {hra}, {other_allow}, {gross}, {itax},  {pf}, {esi}, {odeduct},
{tot_deduc}, {netsal})");
        self.con.commit();


    def salary_entry(self):
        cur = self.con.cursor();
          # gettting percentages for da and hra
        cur.execute("select * from setter;")
        da_per= 0
        hra_per = 0
        for row in cur:
            da_per = row[0];
            hra_per = row[1];

        year = int(input("Enter the year: "))
        month = int(input("Enter the month: "))
        cur.execute("select ecode, basic, TA from emp;")

        ls = [];


        for row in cur:
            ecode = row[0];
            basic = row[1];
            TA = row[2];
            print(f"Details for Ecode:{ecode}")
            no_of_day = int(input("Enter the number of days worked: "))
            da = basic * da_per/100;
            data = da + TA;
            hra = basic * hra_per/100;
            other_allow = int(input("Enter other allowances: "));
            gross = (basic)*(no_of_day/30) + data + hra + other_allow;
            itax = 12/100 * gross;
            pf = 12/100 * basic;
            esi = 1.75/100 * gross;
            otherDeductions = int(input("Enter other deductions: "));
            totalDeductions = itax + pf + esi + otherDeductions;
            netsal = gross - totalDeductions;

            ls.append([year, month, ecode, no_of_day, da, data, hra, other_allow,
gross, itax, pf, esi, otherDeductions, totalDeductions, netsal]);
            print()
            print()



        for smallList in ls:
```

```
            self.salary_entryInd(smallList[0], smallList[1], smallList[2],
smallList[3], smallList[4], smallList[5], smallList[6], smallList[7], smallList[8],
smallList[9], smallList[10] ,smallList[11], smallList[12], smallList[13],
smallList[14]);

        print("Salary Updated successfully..")


    def show_payBill(self, year):
        cur = self.con.cursor();
        cur.execute(f"select * from pay where year = {year}")

        for row in cur:
            print();
            print(f"Year: {row[0]}")
            print(f"Month: {row[1]}")
            print(f"ECode: {row[2]}")
            print(f"No of Days: {row[3]}")
            print(f"DA: {row[4]}")
            print(f"HRA: {row[5]}")
            print(f"Other Allowance: {row[6]}")
            print(f"Gross: {row[7]}")
            print(f"ITax: {row[8]}")
            print(f"PF: {row[9]}")
            print(f"ESI: {row[10]}")
            print(f"Other Deductions: {row[11]}")
            print(f"Total Deductions: {row[12]}")
            print(f"Net Salary: {row[13]}")
            print()
            print();

    def removeEmployee(self, ecode):
        cur = self.con.cursor();
        cur.execute(f"delete from pay where ecode = {ecode}");
        cur.execute(f"delete from emp where ecode = {ecode}");
        self.con.commit();
        print("Data deleted successfully..");




from DBHelper import DBHelper;
import time

def main():
    helper = DBHelper();

    print("*****************************Welcome*********************************")

    while (True):
        print("\nenter 0: to add employee details: ")
        print("\nenter 1: show employee details: ")
```

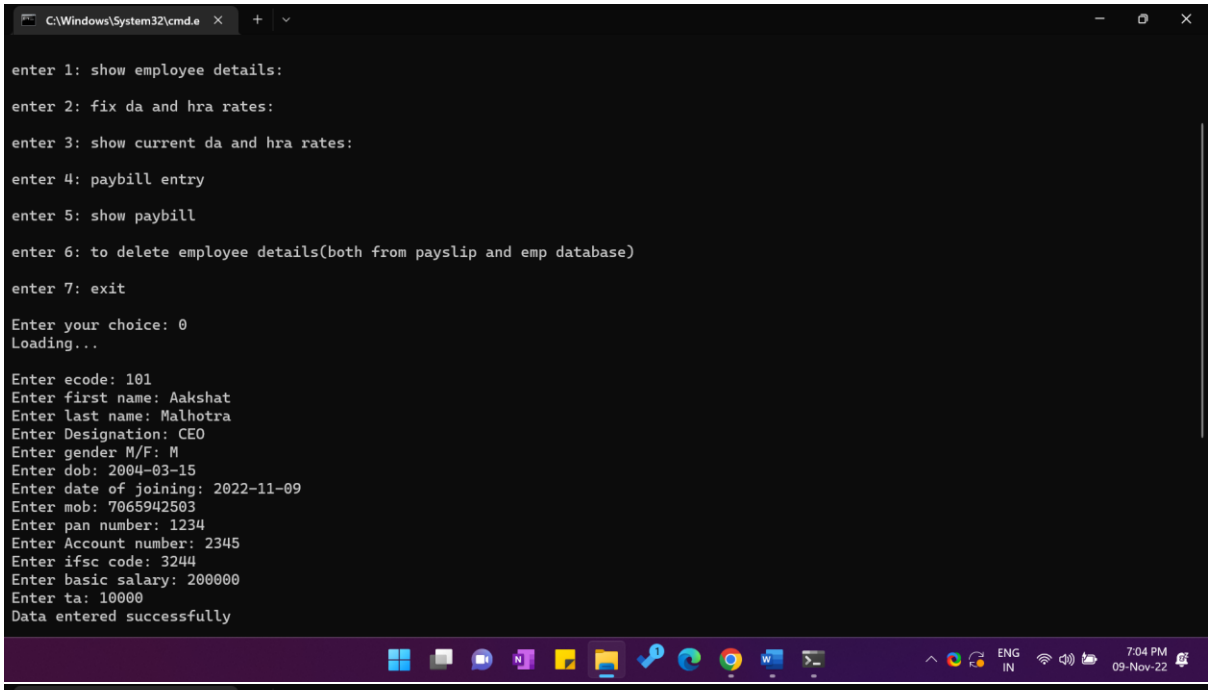
```
        print("\nenter 2: fix da and hra rates: ")
        print("\nenter 3: show current da and hra rates: ")
        print("\nenter 4: paybill entry ")
        print("\nenter 5: show paybill")
        print("\nenter 6: to delete employee details(both from payslip and emp
database)")
        print("\nenter 7: exit\n")

        choic = int(input("Enter your choice: "))
        try:


            if choic == 0:
                print("Loading...\n")
                time.sleep(0.3)
                ecode = int(input("Enter ecode: "));
                fname = input("Enter full name: ");
                lname = input("Enter last name: ");
                desgination = input("Enter Designation: ");
                gender = input("Enter gender M/F: ");
                DOB = input("Enter dob: ");
                DOJ = input("Enter date of joining: ")
                mob = input("Enter mob: ");
                pan = input("Enter pan number: ");
                acno = input("Enter Account number: ");
                ifsc = input("Enter ifsc code: ");
                basic = int(input("Enter basic salary: "));
                ta = int(input("Enter ta: "))
                helper.input_employee(ecode, fname, lname, desgination, gender, DOB,
DOJ, mob, pan, acno, ifsc, basic, ta);

            elif choic == 1:
                print("Loading...\n")
                time.sleep(0.3)
                helper.display_AllEmp();

            elif choic == 2:
                print("Loading...\n")
                time.sleep(0.3)
                daPercent = int(input("Enter da percentage: "))
                hraPercent = int(input("Enter HRA percentage: "))
                helper.percentage_setter(daPercent, hraPercent);

            elif choic == 3:
                print("Loading...\n")
                time.sleep(0.3)
                helper.show_rates();

            elif choic == 4:
                print("Loading...\n")
                time.sleep(0.3)
                helper.salary_entry();

            elif choic == 5:
```

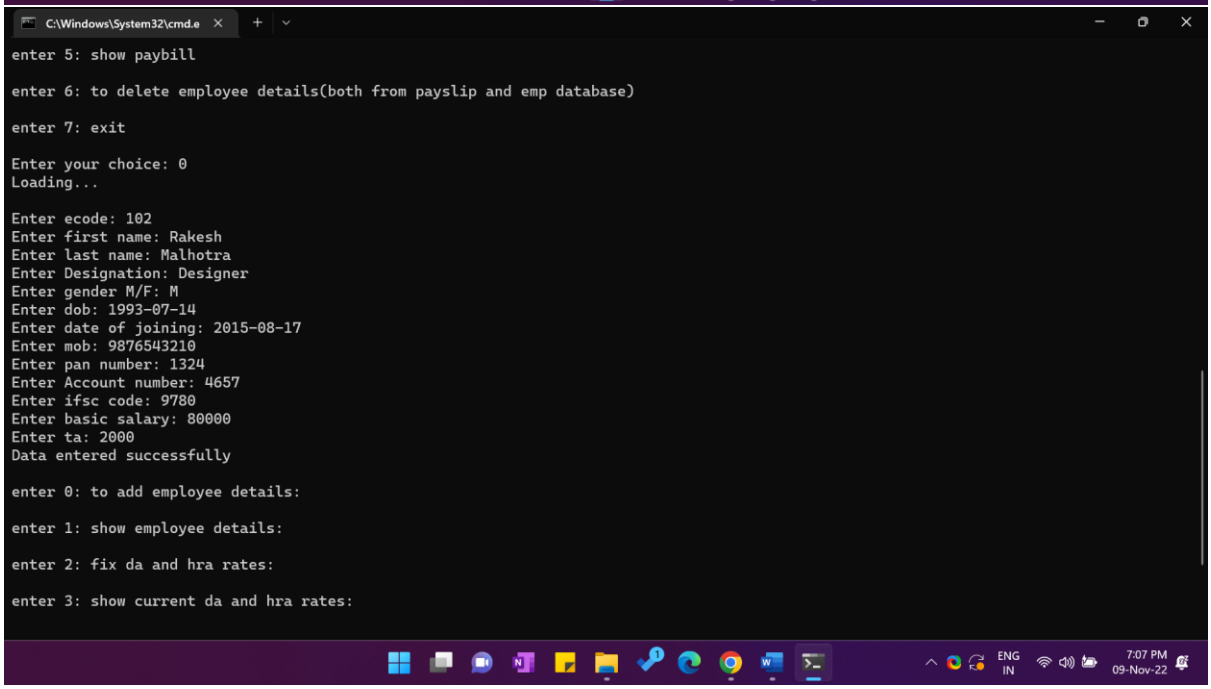
```
                print("Loading...\n")
                time.sleep(0.3)
                year = int(input("Enter the year: "))
                helper.show_payBill(year);

            elif choic == 6:
                print("Loading...\n")
                time.sleep(0.3)
                ecode = int(input("Enter the ecode of the employee whose data u want
to delete: "))
                helper.removeEmployee(ecode);


            elif choic == 7:
                print("Exiting...");
                break;


            else:
                print("Wrong choice.. try again..")


        except Exception as e:
            print(e);



if (__name__ == "__main__"):
    main();
```

Working Screenshots



Entering employee details





Showing Employee Details

```
enter 0: to add employee details:

enter 1: show employee details:

enter 2: fix da and hra rates:

enter 3: show current da and hra rates:

enter 4: paybill entry

enter 5: show paybill

enter 6: to delete employee details(both from payslip and emp database)

enter 7: exit

Enter your choice: 1
Loading...


ECODE: 101
First Name: Aakshat
Last Name: Malhotra
Designation: CEO
Gender: M
DOB: 2004-03-15
DOJ: 2022-11-09
```


```
ECODE: 101
First Name: Aakshat
Last Name: Malhotra
Designation: CEO
Gender: M
DOB: 2004-03-15
DOJ: 2022-11-09
MOB: 7065942503
PAN: 1234
ACNO: 2345
IFSC: 3244
Basic Pay: 200000
TA: 10000


ECODE: 102
First Name: Rakesh
Last Name: Malhotra
Designation: Designer
Gender: M
DOB: 1993-07-14
DOJ: 2015-08-17
MOB: 9876543210
PAN: 1324
ACNO: 4657
IFSC: 9780
Basic Pay: 80000
TA: 2000
```

Setting DA and HRA percentages

```
Enter your choice: 2
Loading...

Enter da percentage: 7
Enter HRA percentage: 6
```

Showing DA and HRA percentage

```
enter 7: exit

Enter your choice: 3
Loading...

DA percentage is 7
HRA percentage is 6
```

Printing paybill for a particular year

```
enter 6: to delete employee details(both from payslip and emp database)

enter 7: exit

Enter your choice: 4
Loading...

Enter the year: 2022
Enter the month: 11
Details for Ecode:101
Enter the number of days worked: 27
Enter other allowances: 10000
Enter other deductions: 3000


Details for Ecode:102
Enter the number of days worked: 20
Enter other allowances: 5000
Enter other deductions: 2000


Salary Updated successfully..
```

```
Enter your choice: 5
Loading...

Enter the year: 2022

Year: 2022
Month: 11
ECode: 101
No of Days: 27
DA: 14000
HRA: 24000
Other Allowance: 12000
Gross: 10000
ITax: 226000
PF: 27120
ESI: 24000
Other Deductions: 3955
Total Deductions: 3000
Net Salary: 58075
```

```
Year: 2022
Month: 11
ECode: 102
No of Days: 20
DA: 5600
HRA: 7600
Other Allowance: 4800
Gross: 5000
ITax: 70733
PF: 8488
ESI: 9600
Other Deductions: 1238
Total Deductions: 2000
Net Salary: 21326
```

Deleting employee details