

ABSTRACT

payroll management is designed to make the existing manual system automatic with the help of computer tools/software, fulfilling their requirement. Manual method to maintain their employee record is difficult and mistakes can happen faster than you think, so the aim of the proposed system is to contribute to the goal of achieving a payroll management system that manages to keep records of employees. It is developed to be used by any organization to manage their data of the employees working in that company. This system is developed to make easy accessibility and can be rearranged as per organization needs so that data can be stored for a longer period.

This project is designed for the purpose of maintaining all details of different employees and their payment and allowances that to be given to the employees of the organization. Employees will be given unique employee ID by the organization, depending on the date of joining and salary date it is created. This program is designed using Python – front end and MySQL – back end. This application can maintain and view computerized record without getting unnecessary entires.

1.INTRODUCTION

1.1. About python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, 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 procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

The Python 2 language, i.e. Python 2.7.x, is "sunsetting" on January 1, 2020 (after extension; first planned for 2015), and the Python team of volunteers will not fix security issues, or improve it in other ways after that date. With the end-of-life, only Python 3.5.x and later will be supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

1.2.About mysql

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

MySQL is based on a client-server model. The core of MySQL is MySQL server, which handles all of the database instructions (or commands). MySQL server is available as a separate program for use in a client-server networked environment and as a library that can be embedded (or linked) into separate applications.

MySQL operates along with several utility programs which support the administration of MySQL databases. Commands are sent to MySQLServer via the MySQL client, which is installed on a computer.

MySQL was originally developed to handle large databases quickly. Although MySQL is typically installed on only one machine, it is able to send the database to multiple locations, as users are able to access it via different MySQL client interfaces. These interfaces send SQL statements to the server and then display the results.

MySQL enables data to be stored and accessed across multiple storage engines, including InnoDB, CSV, and NDB. MySQL is also capable of replicating data and partitioning tables for better performance and durability. MySQL users aren't required to learn new commands; they can access their data using standard SQL commands.

1.3.Overview of Python modules, Tabulate, mysql connector and datetime:

1.3.1.Overview of Python Modules:

The Python interpreter has a number of built-in functions. They are loaded automatically as the interpreter starts and are always available. For example, `print()` and `input()` for I/O, number conversion functions `int()`,`float()`,`complex()`, data type conversions `list()`,`tuple()`, `set()`, etc.

In addition to built-in functions, a large number of pre-defined functions are also available as a part of libraries bundled with Python distributions. These functions are defined in modules. A module is a file containing definitions of functions, classes variables, constants or any other Python objects. Contents of this file can be made available to any other programme.

Built-in modules are written in C and integrated with the Python interpreter. Each builtin module contains resources for certain system-specific functionalities such as OS management, disk IO, etc. The standard library also contains many Python scripts (with the `.py` extension) containing useful utilities.

1.3.2. Overview of Tabulate:

Tabulate is a module that allows you to display table data beautifully. It is not part of standard Python library, so tabulate needs to be installed:

pip install tabulate

Module supports such tabular data types as:

- list of lists (in general case - iterable of iterables).
- dictionary list (or any other iterable object with dictionaries). Keys are used as column names.
- dictionary with iterable objects. Keys are used as column names.

1.3.3. Overview of Mysql.Connector:

MySQL Connector/Python enables Python programs to access MySQL databases, using an API that is compliant with the Python Database API Specification v2.0 (PEP 249). It is written in pure Python and does not have any dependencies except for the Python Standard Library. This module does not come built-in with Python, so mysql.connector needs to be installed:

pip install mysql-connector-python

1.3.3.1 Connecting to MySQL Using Connector/Python

The connect() constructor creates a connection to the MySQL server and returns a MySQLConnection object.

The following example shows how to connect to the MySQL server:

```
import mysql.connector

cnx = mysql.connector.connect(user='root',
password='123456',host='localhost',database='employees')
cnx.close()
```

1.3.4. Overview of DateTime:

In Python, date and time are not a data type of their own, but a module named `datetime` can be imported to work with the date as well as time. Python `Datetime` module comes built into Python, so there is no need to install it externally.

Python `Datetime` module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and `datetime` are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

1.3.4.1. The DateTime module is categorized into 6 main classes :—

- **date**— An idealized naive date, assuming the current Gregorian calendar always was, and always will be, in effect. Its attributes are year, month and day.
- **time** — An idealized time, independent of any particular day, assuming that every day has exactly $24 \times 60 \times 60$ seconds. Its attributes are hour, minute, second, microsecond, and `tzinfo`.
- **datetime** — Its a combination of date and time along with the attributes year, month, day, hour, minute, second, microsecond, and `tzinfo`.
- **timedelta** — A duration expressing the difference between two date, time, or `datetime` instances to microsecond resolution.
- **tzinfo** — It provides time zone information objects.
- **timezone** — A class that implements the `tzinfo` abstract base class as a fixed offset from the UTC

2.SYSTEM REQUIREMENTS

- Processor: Dual Core 1.4GHz Processor
- RAM: Minimum 1 GB
- Hard Disk Space: Minimum 750MB space required.
- Software: Python 3.7.0, MySQL 5.5
- Operating System: Windows 7,8 and 10

3.OBJECTIVE OF THE PROJECT

Payroll refers to the amount of compensation that a business has to pay to its employees after a particular period. Some of its objectives are-

- Maintaining employee's hours and payments.
- Prepare tax payments for each paycheck.
- Calculating bonuses and commissions.
- Print payroll reports for the master budget and expense bulge.
- Automate payroll calculations.
- Save time and reduces administrative burden.
- Consolidate and store records more efficiently.

The primary objective is related to sales, strategy, revenue, etc. while the secondary is associated with the daily tasks of the business.

The aim behind having a payroll management system is to automate and streamline micro tasks such that the HR team to focus on the macro tasks.

Payroll is one of the most important expenses to an organisation, and how its managed plays huge role in the success of your business. A payroll system and how it's used plays a vital role in areas like compliance, employee satisfaction and overall productivity in the organisation. With a cloud-based payroll management system that is consistent with your organisational standards, payroll processing can be made easy and effective.

4.MODULES PURPOSES AND CODING

4.1 Modules Used

- Mysql.connector
- Datetime
- Tabulate

4.2 Modules Purposes

- Mysql.connector : This module is a standardized database driver provided by MySQL.
- Datetime : This module supplies classes for manipulating dates and times.
- Tabulate : This module is an open source python package which is used to print tabular data in nicely formatted tables.

4.3 CODING:

```
#PROJECT
```

```
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='123456')

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 succesfully...")
mycursor.execute(query)

while True:
    print('\n\n\n')
    print("*"*100)
    print('\t\t\t\t\tMAIN MENU')
    print("*"*100)
    print('\t\t\t\t\t1. Adding Employee records')
    print('\t\t\t\t\t2. For Displaying Record of all the
Employees')
    print('\t\t\t\t\t3. For Displaying Record of a particular
Employee')
    print('\t\t\t\t\t4. For Deleting Records of all the
Employees')
    print('\t\t\t\t\t5. For Deleting a Record of a particular
Employee')
    print('\t\t\t\t\t6. For Modification in a record')
    print('\t\t\t\t\t7. For Displaying Payroll')
    print('\t\t\t\t\t8. For Displaying Salary Slip for all
the Employees')
    print('\t\t\t\t\t9. For Displaying Salary Slip for a
particular Employee')
    print('\t\t\t\t\t10. For Exit')
    print('Enter Choice...',end='')
    choice=int(input())

    if choice==1:
        try:
            while True:
                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*0.2
        elif mjob.upper()=='MANAGER':
            mda=mbasic*0.07
            mhra=mbasic*0.30
            mtax=mbasic*0.15
        else:
            mda=mbasic*0.03
            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)

    a=input('do you want to add more record?
    (y/n) ')

    if a.lower()=='n':
        break

    mydb.commit()

    print('Record Added Successfully...')

except:

```

```

        print('Something went wrong')

elif choice==2:
    try:
        query='select * from '+TableName
        mycursor.execute(query)
        #print(query)

print(tabulate(mycursor,headers=['EmpNo','Name','Job','BasicSalary','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)
        myrecord=mycursor.fetchone()
        print("\n\nRecord of Employee No.:"+en)
        print(myrecord)
        c=mycursor.rowcount
        if c== -1:
            print('Nothing to Display')
    except:
        print('Something went wrong')

```

```

elif choice==4:
    try:
        ch=input('Do you want to delete all the records
(y/n) ')
        if ch.upper()=='Y':
            mycursor.execute('delete from '+TableName)
            mydb.commit()
            print('All the records are deleted...')
    except:
        print('Something 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
        #ROWCOUNT is a system variable that is
        #used to return the number of rows that are
        affected by the last executed statement in the batch.
        '''which reflects the number of rows deleted
        #by the executed statement, is retrieved in the
        same execution of the DELETE statement, '''
        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==1:
            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+'''+', '+basicsa
lary='\
            +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(100*'*')
        print('Employee Payroll'.center(90))
        print(100*'*')
        now = datetime.datetime.now()

```

```

        print("Current Date and Time:",end=' ')
        print(now.strftime("%Y-%m-%d %H:%M:%S"))
        print()
        print(100*'-')
        print('%-5s %-15s %-10s %-8s %-8s %-8s %-9s %-
8s %-9s'\

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

        print(100*'-')
        for rec in myrecords:
            print('%4d %-15s %-10s %8.2f %8.2f
%8.2f %9.2f %8.2f %9.2f'%rec)
        print(100*'-')
    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("-"*100)
        print("\t\t\t\tSalary Slip")
        print("-"*100)
        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 %8.2f
%9.2f %8.2f %9.2f'%rec)

            print("-"*100)

        except:

            print('Something went wrong')

    elif choice==9:

        try:

            en=input("Enter employee number whose pay
            slip you want to retrieve:")

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

            mycursor.execute(query)

            now = datetime.datetime.now()

            print("-"*100)

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

            print("-"*100)

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

            ')

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

            print("-"*100)

            #print(query)

            print(tabulate(mycursor,
            headers=['EmpNo', 'Name', 'Job', 'Basic
            Salary', 'DA', 'HRA', 'Gross Salary', 'Tax', 'Net
            Salary'],tablefmt='fancy_grid'))

        except:

            print('Something went wrong')

    elif choice==10:

```

```
        break

    else:
        print('Something went wrong')
```

5. RESULT AND DISCUSSION

5.1 Results

To create a database and table:

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
```

```
>>>
```

```
===== RESTART: C:\Users\ANU\Downloads\project editeed.py =====
```

```
Enter name of your database :-Payroll
```

```
Database created successfully...
```

```
Name of Table to be created:-Pay
```

```
Table Pay created succesfully...
```

Main Menu:

```
*****
```

```
MAIN MENU
```

```
*****
```

1. Adding Employee records
2. For Displaying Record of all the Employees
3. For Displaying Record of a particular Employee
4. For Deleting Records of all the Employees
5. For Deleting a Record of a particular Employee
6. For Modification in a record
7. For Displaying Payroll
8. For Displaying Salary Slip for all the Employees
9. For Displaying Salary Slip for a particular Employee
10. For Exit

```
Enter Choice...|
```

Adding records (choice 1):

```
Enter Choice...1
Enter Employee Information...
Enter Employee no:101
Enter Employee Name:Anuchand
Enter Employee Job:Officer
Enter Basic Salary:50000
do you want to add more record? (y/n)y
Record Added Successfully...
Enter Employee Information...
Enter Employee no:102
Enter Employee Name:Kishor
Enter Employee Job:Manager
Enter Basic Salary:65000
do you want to add more record? (y/n)y
Record Added Successfully...
Enter Employee Information...
Enter Employee no:103
Enter Employee Name:Sanjay
Enter Employee Job:Clerk
Enter Basic Salary:25000
do you want to add more record? (y/n)n
```

For displaying Record for all the employee (Choice 2):

Enter Choice...2

EmpNo	Name	Job	Basic Salary	DA	HRA	Gross Salary	Tax	Net Salary
101	Anuchand	Officer	50000	2500	17500	70000	10000	60000
102	Kishor	Manager	65000	4550	19500	89050	9750	79300
103	Sanjay	Clerk	25000	750	6250	32000	2500	29500

For displaying Record for particular employee (Choice 3):

Enter Choice...3

Enter Employee no. of the record to be displayed...101

Record of Employee No.:101

(101, 'Anuchand', 'Officer', 50000, 2500.0, 17500.0, 70000.0, 10000.0, 60000.0)

For Deleting Records of all the Employees (Choice 4):

Enter Choice...4

Do you want to delete all the records (y/n)y
All the records are deleted...

Enter Choice...2

EmpNo	Name	Job	Basic Salary	DA	HRA	Gross Salary	Tax	Net Salary

For Deleting a Record of a particular Employee (Choice 5):

Enter Choice...5

Enter employee no. of the record to be deleted...103

Deletion done

Enter Choice...2

EmpNo	Name	Job	Basic Salary	DA	HRA	Gross Salary	Tax	Net Salary
101	Anuchand	Officer	50000	2500	17500	70000	10000	60000
102	Kishor	Manager	65000	4550	19500	89050	9750	79300

For Modification in a record (Choice 6):

Enter Choice...6

Enter employee no.of the record to be modified...103

empno : 103

name : Sanjay

job : Clerk

basic : 25000

da : 750.0

hra : 6250.0

gross : 32000.0

tax : 2500.0

net : 29500.0

Type Value to modify below or just press Enter for no change:

Enter name:Arjun

Enter job:

Enter Basic Salary:

update Pay set name='Arjun',job='Clerk',basicsalary=25000 where empno=103

Record modified

For Displaying Payroll (Choice 7):

Enter Choice...7

Employee Payroll

Current Date and Time: 2022-02-12 10:09:57

Empno	Name	Job	Basic	DA	HRA	Gross	Tax	Net
101	Anuchand	Officer	50000.00	2500.00	17500.00	70000.00	10000.00	60000.00
102	Kishor	Manager	65000.00	4550.00	19500.00	89050.00	9750.00	79300.00
103	Arjun	Clerk	25000.00	750.00	6250.00	32000.00	2500.00	29500.00

For Displaying Salary Slip for all the Employees (Choice 8):

Enter Choice...8

Salary Slip								
Current Date and Time: 2022-02-12 10:11:58								
101	Anuchand	Officer	50000.00	2500.00	17500.00	70000.00	10000.00	60000.00
102	Kishor	Manager	65000.00	4550.00	19500.00	89050.00	9750.00	79300.00
103	Arjun	Clerk	25000.00	750.00	6250.00	32000.00	2500.00	29500.00

For Displaying Salary Slip for a particular Employee (Choice 9):

Enter Choice...9

Enter employee number whose pay slip you want to retrieve:102

Salary Slip

Current Date and Time: 2022-02-12 10:13:54

EmpNo	Name	Job	Basic Salary	DA	HRA	Gross Salary	Tax	Net Salary
102	Kishor	Manager	65000	4550	19500	89050	9750	79300

For exit (Choice 10):

MAIN MENU

1. Adding Employee records
2. For Displaying Record of all the Employees
3. For Displaying Record of a particular Employee
4. For Deleting Records of all the Employees
5. For Deleting a Record of a particular Employee
6. For Modification in a record
7. For Displaying Payroll
8. For Displaying Salary Slip for all the Employees
9. For Displaying Salary Slip for a particular Employee
10. For Exit

Enter Choice...10

>>> |

5.2.Discussion

Advantages of payroll software

- work out payroll calculations and deductions quicker
- generate accurate payslips
- calculate bonuses, expenses, holiday pay, etc with minimum effort
- send returns to HMRC and print P45, P60 and other forms for employees
- automate certain tasks, such as year-end reporting
- reduce the burden of compliance
- remove the need to understand complex tax legislation
- store data such as payslips and annual reports in a secure, easily accessible system

Disadvantages of payroll software

- data security, loss or theft
- cyber security and fraud
- information access, quality and control

6.CONCULSION AND FUTURE ENHANCEMENT

6.1 Conclusion

“Payroll Management System” software developed for a company has been designed to achieve maximum efficiency and reduce the time taken to handle the payroll activity. It is designed to replace an existing manual record system thereby reducing time taken for calculations and for storing data. The system uses Python as front end and MySQL as a backend for the database.

This system is strong enough to withstand regressive daily operations under conditions where the database is maintained and cleared over a certain time of span. The implementation of the system in the organisation reduce data entry, time and also provide readily calculated reports.

6.2 Future Enhancement

This project has many future applications like it can be used in any of the Retail Outlet of any type of companies.

This project was build keeping in mind all the requirements of these outlets and they can be implemented in any such type of organisation with very few modification.

7.BIBLIOGRAPHY

1. www.w3school.in
2. www.youtube.com
3. Computer Science with Python, Class XII, Sumita Arora
4. en.wikipedia.org