



Payroll Management System

-Shrutika Mahajan



Overview:

Managing payroll manually can be time-consuming and error-prone. This project aims to simplify payroll operations using a Python-based command-line application connected to a MySQLdatabase

Goal:

To automate salary calculations, streamline record management, and provide flexible reporting features.

Database for storing payroll records:

The screenshot displays a database management interface. On the left, the 'Navigator' pane shows a tree structure under 'SCHEMAS'. The 'organization' schema is expanded, showing 'Tables' with 'employees' and 'payroll'. The 'payroll' table is selected, and its details are shown below: 'Table: payroll' and 'Columns: id (int AI PK), name (varchar(100)), department (varchar(100)), basic_salary (decimal(10,2)), allowances (decimal(10,2)), deductions (decimal(10,2)), net_pay (decimal(10,2))'. The main area shows 'Query 1' with the following SQL code:

```
1 • create database organization;
2
3 • use organization;
4
5 • CREATE TABLE payroll (
6     id INT AUTO_INCREMENT PRIMARY KEY,
7     name VARCHAR(100),
8     department VARCHAR(100),
9     basic_salary DECIMAL(10,2),
10    allowances DECIMAL(10,2),
11    deductions DECIMAL(10,2),
12    net_pay DECIMAL(10,2)
13 );
14
```

The query editor includes a toolbar with icons for file operations, execution, and a 'Limit to 1000 rows' option.

Code:

```
main.py x
1 import csv
2 import pymysql
3
4 # Connect to MySQL
5 try:
6     conn = pymysql.connect(
7         host="localhost",
8         user="root",
9         password="pass@word1",
10        database="organization"
11    )
12    print("Connected successfully!")
13    cursor = conn.cursor()
14 except pymysql.MySQLError as err:
15     print(" Connection failed:", err)
16     exit()
17
18
19 def show_payroll(): 1 usage
20     cursor.execute("SELECT * FROM payroll")
21     for row in cursor.fetchall():
22         print(f"ID: {row[0]}, Name: {row[1]}, Dept: {row[2]}, Basic: {row[3]}, Allowances: {row[4]}, Deductions: {row[5]}")
23
24
```

main.py x

```
25 def insert_payroll(): 1 usage
26     name = input("Enter name: ")
27     department = input("Enter department: ")
28     basic_salary = float(input("Enter basic salary: "))
29     allowances = float(input("Enter allowances: "))
30     deductions = float(input("Enter deductions: "))
31     net_pay = basic_salary + allowances - deductions
32     cursor.execute(
33         query: "INSERT INTO payroll (name, department, basic_salary, allowances, deductions, net_pay) VA
34         args: (name, department, basic_salary, allowances, deductions, net_pay)
35     )
36     conn.commit()
37     print("Payroll record added.")
38
39 def update_payroll(): 1 usage
40     emp_id = int(input("Enter payroll ID to update: "))
41     name = input("Enter new name: ")
42     department = input("Enter new department: ")
43     basic_salary = float(input("Enter new basic salary: "))
44     allowances = float(input("Enter new allowances: "))
45     deductions = float(input("Enter new deductions: "))
46     net_pay = basic_salary + allowances - deductions
47     cursor.execute(
48         query: "UPDATE payroll SET name=%s, department=%s, basic_salary=%s, allowances=%s, deductions=%s
49         args: (name, department, basic_salary, allowances, deductions, net_pay, emp_id)
50     )
51     conn.commit()
52     print("Payroll record updated.")
```

Activate Windows

main.py x

```
55
56 def delete_payroll(): 1 usage
57     emp_id = int(input("Enter payroll ID to delete: "))
58     cursor.execute(query: "DELETE FROM payroll WHERE id=%s", args: (emp_id,))
59     conn.commit()
60     print(" Payroll record deleted.")
61 def export_to_csv(): 1 usage
62     cursor.execute("SELECT * FROM payroll")
63     rows = cursor.fetchall()
64     with open('payroll.csv', 'w', newline='') as file:
65         writer = csv.writer(file)
66         writer.writerow(['ID', 'Name', 'Department', 'Basic Salary', 'Allowances', 'Deductions', 'Net Pay'])
67         writer.writerows(rows)
68     print(" Data exported to payroll.csv")
69
70
71 def search_payroll_by_name(): 1 usage
72     name = input("Enter name to search: ")
73     cursor.execute(query: "SELECT * FROM payroll WHERE name LIKE %s", args: ('%' + name + '%',))
74     results = cursor.fetchall()
75     if results:
76         for row in results:
77             print(f"ID: {row[0]}, Name: {row[1]}, Dept: {row[2]}, Basic: {row[3]}, Allowances: {row[4]}, Deducti
78     else:
79         print("No matching records found.")
80
```

main.py x

```
80
81
82 def payroll_summary_by_department(): 1 usage
83     cursor.execute("""
84         SELECT department, COUNT(*), SUM(basic_salary), SUM(allowances), SUM(deductions), SUM(net_pay)
85         FROM payroll
86         GROUP BY department
87     """)
88     for row in cursor.fetchall():
89         print(f"Dept: {row[0]}, Employees: {row[1]}, Total Basic: {row[2]}, Total Allowances: {row[3]}, Total Deductions: {row[4]}, Total Net Pay: {row[5]}")
90
91
92 def filter_by_salary_range(): 1 usage
93     min_salary = float(input("Enter minimum net pay: "))
94     max_salary = float(input("Enter maximum net pay: "))
95     cursor.execute(query="SELECT * FROM payroll WHERE net_pay BETWEEN %s AND %s", args=(min_salary, max_salary))
96     results = cursor.fetchall()
97     if results:
98         for row in results:
99             print(f"ID: {row[0]}, Name: {row[1]}, Dept: {row[2]}, Net Pay: {row[6]}")
100     else:
101         print("No employees found in this salary range.")
102
```


main.py x

```
104     def menu(): 1 usage
105         while True:
106             print("\n MENU")
107             print("1. Show Payroll")
108             print("2. Insert Payroll")
109             print("3. Update Payroll")
110             print("4. Delete Payroll")
111             print("5. Export to CSV")
112             print("6. Search by Name")
113             print("7. Summary by Department")
114             print("8. Filter by Salary Range")
115             print("9. Exit")
116             choice = input("Enter choice: ")
117             if choice == '1':
118                 show_payroll()
119             elif choice == '2':
120                 insert_payroll()
121             elif choice == '3':
122                 update_payroll()
123             elif choice == '4':
124                 delete_payroll()
125             elif choice == '5':
126                 export_to_csv()
127             elif choice == '6':
128                 search_payroll_by_name()
129             elif choice == '7':
130                 payroll_summary_by_department()
131             elif choice == '8':
132                 filter_payroll_by_salary_range()
```




Output:

MENU

1. Show Payroll
2. Insert Payroll
3. Update Payroll
4. Delete Payroll
5. Export to CSV
6. Search by Name
7. Summary by Department
8. Filter by Salary Range
9. Exit

Enter choice: 1

ID: 1, Name: shrutika, Dept: it, Basic: 50000000.00, Allowances: 5000000.00, Deductions: 500.00, Net Pay: 54999500.00

ID: 2, Name: abc, Dept: it, Basic: 78657.00, Allowances: 764.00, Deductions: 67.00, Net Pay: 79354.00

MENU

1. Show Payroll
2. Insert Payroll
3. Update Payroll
4. Delete Payroll
5. Export to CSV
6. Search by Name
7. Summary by Department
8. Filter by Salary Range
9. Exit

Enter choice: 2

Enter name: shruti

Enter department: it

Enter basic salary: 7000000

Enter allowances: 8000000

Enter deductions: 67

Payroll record added.

MENU

1. Show Payroll
2. Insert Payroll
3. Update Payroll
4. Delete Payroll
5. Export to CSV
6. Search by Name
7. Summary by Department
8. Filter by Salary Range
9. Exit

Enter choice: 3

Enter payroll ID to update: 3

Enter new name: shrutii

Enter new department: it

Enter new basic salary: 8000000

Enter new allowances: 7000000

Enter new deductions: 78

Payroll record updated.

```
4. Delete Payroll
5. Export to CSV
6. Search by Name
7. Summary by Department
8. Filter by Salary Range
9. Exit
Enter choice: 4
Enter payroll ID to delete: 3
Payroll record deleted.
```

MENU

```
1. Show Payroll
2. Insert Payroll
3. Update Payroll
4. Delete Payroll
5. Export to CSV
6. Search by Name
7. Summary by Department
8. Filter by Salary Range
9. Exit
```

Enter choice: 1

ID: 1, Name: shrutika, Dept: it, Basic: 50000000.00, Allowances: 5000000.00, Deductions: 500.00, Net Pay: 54999500.00

ID: 2, Name: abc, Dept: it, Basic: 78657.00, Allowances: 764.00, Deductions: 67.00, Net Pay: 79354.00

PayrollManagement C:\Use

> .venv library root

main.py

payroll.csv

External Libraries

Scratches and Consoles

104

121

122

123

124

125

126

127

def menu(): 1 usage

elif choice == '3':

update_payroll()

elif choice == '4':

delete_payroll()

elif choice == '5':

export_to_csv()

elif choice == '6':

Run

main x

↺

⏏

⋮

8. Filter by salary range

9. Exit

Enter choice: 5

Data exported to payroll.csv

MENU

1. Show Payroll

2. Insert Payroll

6. Search by Name

7. Summary by Department

8. Filter by Salary Range

9. Exit

Enter choice: 6

Enter name to search: shrutika

ID: 1, Name: shrutika, Dept: it, Basic: 50000000.00, Allowances: 5000000.00, Deductions: 500.00, Net Pay: 54999500.00

MENU

1. Show Payroll

2. Insert Payroll

3. Update Payroll

4. Delete Payroll

5. Export to CSV

6. Search by Name

7. Summary by Department

8. Filter by Salary Range

9. Exit

Enter choice: 7

Dept: it, Employees: 2, Total Basic: 50078657.00, Total Allowances: 5000764.00, Total Deductions: 567.00, Total Net Pay: 55078854.00

6. Search by Name
7. Summary by Department
8. Filter by Salary Range
9. Exit

Enter choice: 8

Enter minimum net pay: 1000

Enter maximum net pay: 10000000000

ID: 1, Name: shrutika, Dept: it, Net Pay: 54999500.00

ID: 2, Name: abc, Dept: it, Net Pay: 79354.00

MENU

1. Show Payroll
2. Insert Payroll
3. Update Payroll
4. Delete Payroll
5. Export to CSV
6. Search by Name
7. Summary by Department
8. Filter by Salary Range
9. Exit

Enter choice: 9

Goodbye!