

Employee Management System

- 1) Overview & MySQL installation
- 2) Learn SQL & Database Structure
- 3) MySQL with Python & Features development
- 4) Frontend Development
- 5) Feature Integration in Frontend

Overview

What

- 1) The Employee Management System is a web-based application designed to streamline employee data management, including personal information, job roles, salaries, attendance, and performance.
- 2) It offers functionalities such as viewing employee records, adding new employees, updating details, tracking attendance, managing payroll, and generating reports.
- 3) MySQL is used as the database management system (DBMS) to store and manage all employee-related data efficiently.
- 4) The system is designed for deployment within an organization's local area network (LAN) and can be accessed securely.

Why

- 1) Organizations, HR departments, and administrative teams benefit from an automated system that eliminates manual record-keeping and enhances efficiency.
- 2) This project demonstrates expertise in backend database management, SQL queries, and project development capabilities, making it a valuable addition to a developer's portfolio.
- 3) It reduces errors, improves decision-making through data analytics, and streamlines HR operations.

How

- 1) Backend: MySQL, SQL, Python
- 2) Frontend: Stream lit for user-friendly web-based interactions

Database Structure

Company

Employees	Departments	Salaries	Attendance	Performance
emp_id(P)	dept_id(P)	salary_id(P)	attndnc_id(P)	perfmnce_id(P)
first_name	dept_name	emp_id	emp_id	emp_id
last_name		base_sal	date	evaltion_date
date of birth		bonuses	check_in	rating
gender		deductions	check_out	feedback
phn_num		net_salary		
email				
address				
hire_date				
job_role				
dept_id				
salary_id				
status				

SQL: Structured Query Language

1.Create a Database

```
Create database database_name;
```

2.Delete a Database

```
drop database database_name;
```

3.Use a Database

```
use database_name;
```

4.To create a table

```
Create table table_name(column_name1 datatype(size),  
column_name2 datatype(size),...., primary_key(column_name1));
```

5.To insert values into table

```
Insert into table_name values(column1_value, column2_value,....);
```

6.To retrieve data from table

```
select * from table_name;
```

7.To modify data

Update table_name

set column_name=new_value

where primary_key=value;

8.To delete data from table

Delete from table_name

where primary_key=value;

9.To delete table

drop table table_name;

The screenshot shows the MySQL Workbench interface. The Navigator pane on the left displays the 'company' schema with its tables: 'employees' and 'departments'. The 'employees' table is selected, showing its columns: employee_id, first_name, last_name, date_of_birth, gender, phone_number, email, address, hire_date, job_role, department_id, salary_id, and status. The SQL Editor tab contains the following query:

```
1 *  select * from employees;
2
```

The Result Grid shows the following data:

employee_id	first_name	last_name	date_of_birth	gender	phone_number	email	address	hire_date	job_role	department_id	salary_id	status
1	Anu	PS	1992-06-13	F	8534685632	anu@gmail.com	Anu vihar,Bangalore	2023-02-10	Data Analyst	1	5	Active
2	Kevin	Jacob	1990-06-15	M	7956993486	kevin@gmail.com	kevin vihar,Kerala	2022-05-17	HR	2	4	Active

SCHEMAS

Filter objects

company

- Tables
 - employees
- Columns
 - employee_id
 - first_name
 - last_name
 - date_of_birth
 - gender
 - phone_number
 - email
 - address
 - hire_date
 - job_role
 - department_id

Administration Schemas

Information:

Schema: company

```

1 • create table departments(dept_id varchar(255),dept_name varchar(255),primary key(dept_id));
2 • select * from employees;
3 • insert into departments values("4","Developer");
4 • select * from departments;

```

Result Grid

dept_id	dept_name
1	Data Analyst
2	HR
3	Data Scientist
4	Developer
*	HULL

Result Grid Form Editor Field Types

SCHEMAS

Filter objects

company

- Tables
 - employees
- Columns
 - employee_id
 - first_name
 - last_name
 - date_of_birth
 - gender
 - phone_number
 - email
 - address
 - hire_date
 - job_role
 - department_id

Administration Schemas

Information:

Schema: company

```

2 • select * from employees;
3 • insert into departments values("4","Developer");
4 • select * from departments;
5 • create table salaries(sal_id varchar(255),employee_id varchar(255),base_salary DECIMAL(10,2),b
6 • insert into salaries values("2","2","50000","4000","1000","53000");
7 • select * from salaries;

```

Result Grid

sal_id	employee_id	base_salary	bonuses	deductions	net_salary
1	1	60000.00	3000.00	1000.00	62000.00
2	2	50000.00	4000.00	1000.00	53000.00
*	HULL	HULL	HULL	HULL	HULL

Result Grid Form Editor Field Types

SCHEMAS

Filter objects

company

- Tables
 - employees
- Columns
 - employee_id
 - first_name
 - last_name
 - date_of_birth
 - gender
 - phone_number
 - email
 - address
 - hire_date
 - job_role
 - department_id

Administration Schemas

Information:

Schema: company

```

5 • create table salaries(sal_id varchar(255),employee_id varchar(255),base_salary DECIMAL(10,2),b
6 • insert into salaries values("2","2","50000","4000","1000","53000");
7 • select * from salaries;
8 • create table attendance(attendance_id varchar(255),employee_id varchar(255),date DATE,check_in
9 • insert into attendance values("2","2","2024-09-18","9.10","18.30","Present");
10 • select * from attendance;

```

Result Grid

attendance_id	employee_id	date	check_in	check_out	status
1	1	2024-09-18	00:00:09	00:00:18	Present
2	2	2024-09-18	00:00:09	00:00:18	Present
*	HULL	HULL	HULL	HULL	HULL

Result Grid Form Editor Field Types

Navigator

Schemas

Filter objects

company

- Tables
 - employees
 - Columns
 - employee_id
 - first_name
 - last_name
 - date_of_birth
 - gender
 - phone_number
 - email
 - address
 - hire_date
 - job_role
 - department_id

Administration Schemas

Information

Schema: company

SQL File 3

```

9 • insert into attendance values("2","2","2024-09-18","9.10","18.30","Present");
10 • select * from attendance;
11 • create table performance(performance_id varchar(255),employee_id varchar(255),evaluation_date DATE,rating DECIMAL(2,1) CHECK (rating BETWEEN 1 AND 5),feedback TEXT,primary key(performance_id));
12 • insert into performance values("2","2","2024-11-11","3.5","Good");
13 • drop table performance;
14 • select * from performances;

```

Result Grid

performance_id	employee_id	evaluation_date	rating	feedback
1	1	2024-11-11	4.5	Excellent
2	2	2024-11-11	3.5	Good
*	HULL	HULL	HULL	HULL

Form Editor

Field Types

```

1 • create table departments(dept_id varchar(255),dept_name varchar(255),primary key(dept_id));
2 • select * from employees;
3 • insert into departments values("4","Developer");
4 • select * from departments;
5 • create table salaries(sal_id varchar(255),employee_id varchar(255),base_salary DECIMAL(10,2),bonuses DECIMAL(10,2),deductions DECIMAL(10,2),net_salary DECIMAL(10,2),primary key(sal_id));
6 • insert into salaries values("2","","50000","4000","1000","53000");
7 • select * from salaries;
8 • create table attendance(attendance_id varchar(255),employee_id varchar(255),date DATE,check_in TIME,check_out TIME,status ENUM('Present','Absent','Leave'),primary key(attendance_id));
9 • insert into attendance values("2","2","2024-09-18","9.10","18.30","Present");
10 • select * from attendance;
11 • create table performance(performance_id varchar(255),employee_id varchar(255),evaluation_date DATE,rating DECIMAL(2,1) CHECK (rating BETWEEN 1 AND 5),feedback TEXT,primary key(performance_id));
12 • insert into performance values("2","2","2024-11-11","3.5","Good");
13 • drop table performance;
14 • select * from performances;

```

Python

- 1.Create a virtual environment
- 2.Activate the virtual environment every time you use it.
- 3.Install the packages mysql-connector-python

```
import mysql.connector  
  
mydb =  
mysql.connector.connect(host="localhost",user="root",password="1  
234",database="company")
```

Output

```
(employee) C:\projects\employee\Scripts>python backend.py  
<mysql.connector.connection_cext.CMySQLConnection object at 0x000001CF3B08EC90>
```

```
import mysql.connector  
  
mydb =  
mysql.connector.connect(host="localhost",user="root",password="1  
234",database="company")  
  
# To retrieve data from database  
c=mydb.cursor()  
c.execute("select * from employees")  
for r in c:  
    print(r)
```

output

```
(employee) C:\projects\employee\Scripts>python backend.py
('1', 'Anu', 'PS', datetime.date(1992, 6, 13), 'F', '8534685632', 'anu@gmail.com', 'Anu vihar,Bangalore', datetime.date(2023, 2, 10), 'Data Analyst', '1', '5', 'Active')
('2', 'Kevin', 'Jacob', datetime.date(1990, 6, 15), 'M', '7956893486', 'kevin@gmail.com', 'kevin vihar,Kerala', datetime.date(2022, 5, 17), 'HR', '2', '4', 'Active')
```

```
import mysql.connector

mydb =
mysql.connector.connect(host="localhost",user="root",password="1234",database="company")

# To retrieve data from database

c=mydb.cursor()

c.execute("select * from employees")

for r in c:

    print(r[1])
```

```
(employee) C:\projects\employee\Scripts>python backend.py
Anu
Kevin
```

```
import mysql.connector

mydb =
mysql.connector.connect(host="localhost",user="root",password="1234",database="company")

# To retrieve multiple tables from database
```

```
c=mydb.cursor()  
c.execute("select * from employees")  
for r in c:  
    print(r)
```

```
c2=mydb.cursor()  
c2.execute("select * from salaries")  
for r in c2:  
    print(r)
```

```
(employee) C:\projects\employee\Scripts>python backend.py  
('1', 'Anu', 'PS', datetime.date(1992, 6, 13), 'F', '8534685632', 'anu@gmail.com', 'Anu vihar,Bangalore', datetime.date(2023, 2, 10), 'Data Analyst', '1', '5', 'Active')  
('2', 'Kevin', 'Jacob', datetime.date(1990, 6, 15), 'M', '7956893486', 'kevin@gmail.com', 'kevin vihar,Kerala', datetime.date(2022, 5, 17), 'HR', '2', '4', 'Active')  
('1', '1', Decimal('60000.00'), Decimal('3000.00'), Decimal('1000.00'), Decimal('62000.00'))  
('2', '2', Decimal('50000.00'), Decimal('4000.00'), Decimal('1000.00'), Decimal('53000.00'))
```

```
import mysql.connector  
  
import datetime  
  
mydb =  
mysql.connector.connect(host="localhost",user="root",password="1234",database="company")
```

```
#To insert data to database

did=input("enter the department id")

dname=input("enter the department name")

c=mydb.cursor()

sql = "INSERT INTO Departments (dept_id, dept_name)
VALUES (%s, %s)"

values = (did, dname)

c.execute(sql, values)

mydb.commit()

print("department details issued successfully")

c.close()

mydb.close()
```

```
(employee) C:\projects\employee\Scripts>python backend.py
enter the department id5
enter the department namefinance
deoartment details issued successfully
```

Frontend Development

Web Application

- Creating a web application with the help of streamlit.
- pip install streamlit (install streamlit in virtual environment)
- Open IDLE and create a new python file called main.py. Using this python file we can write code create web application.
- We can add title to this page and save the file as main.py.

```
import streamlit as st
import mysql.connector
st.title("EMPLOYEE MANAGEMENT SYSTEM")
```

- Using virtual environment command prompt run streamlit using command: streamlit run main.py

```
(employee) C:\projects\employee\Scripts>streamlit run main.py

Welcome to Streamlit!

If you'd like to receive helpful onboarding emails, news, offers, promotions,
and the occasional swag, please enter your email address below. Otherwise,
leave this field blank.

Email:

You can find our privacy policy at https://streamlit.io/privacy-policy

Summary:
- This open source library collects usage statistics.
- We cannot see and do not store information contained inside Streamlit apps,
such as text, charts, images, etc.
- Telemetry data is stored in servers in the United States.
- If you'd like to opt out, add the following to %UserProfile%/.streamlit/config.toml,
creating that file if necessary:

[browser]
gatherUsageStats = false

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.1.10:8501
```

- Browser will open and we can see the title in the browser.

EMPLOYEE MANAGEMENT SYSTEM

- Also change the title of the project using `st.set_page_config(page_title="Employee Management System",page_icon=" ")`
- Similarly we can add image, text, video, button, sidebar dropdown menu etc to it.



- Also, we can add session into our program. It helps to get different sessions like home, user, employee, department.

- In the user page I add features of user registration, login and logout

```

import streamlit as st    # st used as name streamlit for making interactive web application
import pandas as pd
import mysql.connector
st.title("EMPLOYEE MANAGEMENT SYSTEM")
choice=st.sidebar.selectbox("My Menu",("Home","User","Employee","Department"))
st.write(choice)
if(choice=="Home"):
    st.image("https://juntrax.com/blog/wp-content/uploads/2021/01/Employee-Management-System-1024x585.jpg")
elif(choice=="User"):

    if 'login' not in st.session_state:
        st.session_state['login']=False

    if 'user_id' not in st.session_state:
        st.session_state['user_id'] = None

    menu = st.radio("Choose an option:", ("Login", "Register"))

    #Login

    if menu == "Login":

        uid=st.text_input("Enter UserID")
        upwd=st.text_input("Enter Password")
        btn=st.button("Login")
        if btn:
            mydb=mysql.connector.connect(host="localhost",user="root",password="1234",database="company")
            c=mydb.cursor()
            c.execute("select * from user")
            for r in c:
                if(r[0]==uid and r[1]==upwd):
                    st.session_state['login']=True
                    break

            if(not st.session_state['login']):
                st.write("incorrect ID or Password")

        if(st.session_state['login']):
            st.write("Login succesful")

    # Registration

    elif menu == "Register":
        new_uid = st.text_input("Create userid")
        new_pwd = st.text_input("create password",type="password")
        confirm_pwd = st.text_input("Confirm Password",type="password")
        reg_btn = st.button("Register")

        if reg_btn:
            if new_pwd != confirm_pwd:
                st.error("Password donot match")
            else:
                mydb=mysql.connector.connect(host="localhost",user="root",password="1234",database="company")
                c=mydb.cursor()

                c.execute("SELECT * FROM user WHERE user_id = %s", (new_uid,))
                existing_user = c.fetchone()

                if existing_user:
                    st.error("UserID already exists. Try a different one.")
                else:
                    c.execute("INSERT INTO user (user_id, password) VALUES (%s, %s)", (new_uid, new_pwd))
                    mydb.commit()
                    st.success("Registration successful! Please log in.")

    # Logout

    if st.session_state['login']:
        st.write("Welcome")
        if st.button("Logout"):
            st.session_state['login']=False
            st.session_state['user_id']=None
            st.success("Logged out successfully")

```

- Feature to check password and confirmation password

EMPLOYEE MANAGEMENT SYSTEM

User

Choose an option:

Login
 Register

Create userid

create password

Confirm Password

Register

Password do not match

- Feature to Registration

EMPLOYEE MANAGEMENT SYSTEM

User

Choose an option:

Login
 Register

Create userid

create password

Confirm Password

Register

Registration successful! Please log in.

- Feature to Login

EMPLOYEE MANAGEMENT SYSTEM

User

Choose an option:

- Login
- Register

Enter UserID

4

Enter Password

appu

[Login](#)

Login successful

Welcome

[Logout](#)

- Feature to log out

EMPLOYEE MANAGEMENT SYSTEM

User

Choose an option:

- Login
- Register

Enter UserID

4

Enter Password

appu

[Login](#)

Login successful

Welcome

[Logout](#)

Logged out successfully

- After login we can see the table details of employees, salaries, department, attendance and performance

Login

Login succesful

Welcome

View Employee Management Data

Choose Table to View

Employees

Employees Table

	employee_id	first_name	last_name	date_of_birth	gender	phone_number	email	ac
0	1	Anu	PS	1992-06-13	F	8534685632	anu@gmail.com	Ari
1	2	Kevin	Jacob	1990-06-15	M	7956893486	kevin@gmail.com	ke
2	3	kichu	k	1988-02-16	F			

Logout

- Similar to Home page and User page, I create page to add, update and delete employee's details

```

elif choice == "Employee":
    st.subheader("Employee Management")
    action = st.selectbox("Choose action", ("None", "Add Employee", "Update Employee", "Delete Employee"))

if action == "Add Employee":
    emp_id = st.text_input("Enter Employee ID")
    first_name = st.text_input("Enter First Name")
    last_name = st.text_input("Enter Last Name")
    date_of_birth = st.date_input("Enter Date of Birth")
    gender = st.selectbox("Select Gender", ["M", "F", "Other"])
    phone_number = st.text_input("Enter Phone Number")
    email = st.text_input("Enter Email")
    address = st.text_area("Enter Address")
    hire_date = st.date_input("Enter Hire Date")
    job_role = st.text_input("Enter Job Role")
    department_id = st.text_input("Enter Department ID")
    salary_id = st.text_input("Enter Salary ID")
    status = st.selectbox("Select Status", ["Active", "Inactive"])

    add_btn = st.button("Add Employee")

if add_btn:
    mydb=mysql.connector.connect(host="localhost",user="root",password="1234",database="company")
    c=mydb.cursor()
    c.execute("""
        INSERT INTO employees
        (employee_id,first_name, last_name, date_of_birth, gender, phone_number, email, address, hire_date, job_role, department_id, salary_id, status)
        VALUES (%s,%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
        """,
        (emp_id,first_name, last_name, date_of_birth, gender, phone_number, email, address, hire_date, job_role, department_id, salary_id, status))
    )
    mydb.commit()
    st.success("Employee added successfully!")

```

```

elif action == "Update Employee":
    emp_id = st.text_input("Enter Employee ID to Update")
    update_field = st.selectbox("Select Field to Update", ["employee_id", "first_name", "last_name", "date_of_birth", "gender", "phone_number", "email", "address", "hire_date", "job_role"])
    new_value = st.text_input("Enter New Value")
    update_btn = st.button("Update Employee")

if update_btn:
    mydb = mysql.connector.connect(host="localhost", user="root", password="1234", database="company")
    c = mydb.cursor()
    c.execute(f"UPDATE employees SET {update_field} = %s WHERE employee_id = %s", (new_value, emp_id))
    mydb.commit()
    st.success("Employee updated successfully!")

elif action == "Delete Employee":
    emp_id = st.text_input("Enter Employee ID to Delete")
    delete_btn = st.button("Delete Employee")

if delete_btn:
    if emp_id.strip(): # Ensure emp_id is not empty
        mydb = mysql.connector.connect(host="localhost", user="root", password="1234", database="company")
        c = mydb.cursor()
        c.execute("DELETE FROM employees WHERE employee_id = %s", (emp_id,))
        mydb.commit()
        st.success("Employee deleted successfully!")
    else:
        st.error("Please enter a valid Employee ID!")

```

Employee

Employee Management

Choose action

Add Employee

Enter Employee ID

3

Enter First Name

Greeshma

Enter Last Name

PS

Enter Date of Birth

2025/02/18

Select Gender

M

Enter Phone Number

EMPLOYEE MANAGEMENT SYSTEM

Employee

Employee Management

Choose action

Delete Employee

Enter Employee ID to Delete

4

Delete Employee

Employee deleted successfully!

Employee

Employee Management

Choose action

Update Employee

Enter Employee ID to Update

3

Select Field to Update

employee_id

Enter New Value

1992-06-13

Update Employee

Employee updated successfully!

- Also create a page to add, update & delete employee salary.

```

if action == "Add Salary":
    sal_id=st.text_input("Enter salary ID")
    eid = st.text_input("Enter Employee ID")

    base_salary = st.number_input("Enter base salary",min_value=0.0,format="%2f")
    bonuses=st.number_input("Enter Bonuses(if any)",min_value=0.0,format="%2f")
    deductions=st.number_input("Enter Deductions(if any)",min_value=0.0,format="%2f")

    net_salary=base_salary+bonuses-deductions
    st.write(f"Calculated net salary: {net_salary}")

if st.button("Add Salary"):
    mydb = mysql.connector.connect(host="localhost",user="root",password="1234",database="company")
    c = mydb.cursor()
    c.execute("INSERT INTO salaries (sal_id,employee_id, base_salary, bonuses, deductions, net_salary) VALUES (%s,%s, %s, %s, %s, %s)", (sal_id,eid, base_salary, bonuses, deductions, net_salary))
    mydb.commit()
    c.close()
    mydb.close()
    st.success("Salary added successfully !")

elif action == "Update Salary":
    sal_id = st.text_input("Enter Salary ID")
    update_field=st.selectbox("Select Field to Update",["base_salary", "bonuses", "deductions", "net_salary"])
    new_value=st.number_input("Enter New Value",min_value=0.0,format="%2f")

    if st.button("Update Salary"):
        mydb = mysql.connector.connect(host="localhost",user="root",password="1234",database="company")
        c = mydb.cursor()
        c.execute(f"UPDATE salaries SET {update_field} = %s WHERE sal_id = %s", (new_value, sal_id))
        mydb.commit()
        c.close()
        mydb.close()
        st.success("Salary updated successfully !")

```

Salary Management

Choose action

Add Salary

Enter salary ID

5

Enter Employee ID

4

Enter base salary

20000.00

- +

Enter Bonuses(if any)

2000.00

- +

Enter Deductions(if any)

500.00

- +

Calculated net salary: 21500.0

Add Salary

Salary added successfully !

>

Deploy



EMPLOYEE MANAGEMENT SYSTEM

Salary

Salary Management

Choose action

Update Salary

Enter Salary ID

3

Select Field to Update

bonuses

Enter New Value

6000

- +

Update Salary

Salary updated successfully !

```
elif action == "Delete Salary":  
    sal_id = st.text_input("Enter Salary ID to Delete")  
  
    if st.button("Delete Salary"):  
        if sal_id.strip():  
            mydb = mysql.connector.connect(host="localhost", user="root", password="1234", database="company")  
            c = mydb.cursor()  
            c.execute("DELETE FROM salaries WHERE sal_id = %s", (sal_id,))  
            mydb.commit()  
            c.close()  
            mydb.close()  
            st.success("Salary deleted successfully!")  
        else:  
            st.error("Please enter a valid Sal_ID!")
```

EMPLOYEE MANAGEMENT SYSTEM

Salary

Salary Management

Choose action

Delete Salary

Enter Salary ID to Delete

4

Delete Salary

Salary deleted successfully!

- Then create page to add, update and delete department details.

```

elif(choice=="Department"):
    st.subheader("Department Management")
    dept_action = st.selectbox("Choose action", ("None", "Add Department", "Update Department","Delete Department"))

if dept_action == "Add Department":
    dept_id = st.text_input("Enter Department ID")
    dept_name = st.text_input("Enter Department Name")
    add_dept_btn = st.button("Add Department")

    if add_dept_btn:
        mydb = mysql.connector.connect(host="localhost",user="root",password="1234",database="company")
        c = mydb.cursor()
        c.execute("INSERT INTO departments (dept_id, dept_name) VALUES (%s, %s)", (dept_id, dept_name))
        mydb.commit()
        c.close()
        mydb.close()
        st.success("Department added successfully!")

elif dept_action == "Update Department":
    dept_id = st.text_input("Enter Department ID to Update")
    new_dept_name = st.text_input("Enter New Department Name")
    update_dept_btn = st.button("Update Department")

    if update_dept_btn:
        mydb = mysql.connector.connect(host="localhost",user="root",password="1234",database="company")
        c = mydb.cursor()
        c.execute("UPDATE departments SET dept_name = %s WHERE dept_id = %s", (new_dept_name, dept_id))
        mydb.commit()
        c.close()
        mydb.close()
        st.success("Department updated successfully!")

elif dept_action == "Delete Department":
    dept_id = st.text_input("Enter Department ID to Delete")
    delete_dept_btn = st.button("Delete Department")

    if delete_dept_btn:
        mydb = mysql.connector.connect(host="localhost",user="root",password="1234",database="company")
        c = mydb.cursor()
        c.execute("DELETE FROM departments WHERE dept_id = %s", (dept_id,))
        mydb.commit()
        c.close()
        mydb.close()
        st.success("Department deleted successfully!")

```

EMPLOYEE MANAGEMENT SYSTEM

Department

Department Management

Choose action

Add Department

Enter Department ID

8

Enter Department Name

Admin

Add Department

Department added successfully!

EMPLOYEE MANAGEMENT SYSTEM

Department

Department Management

Choose action

Update Department

Enter Department ID to Update

1

Enter New Department Name

Data Scientist

Update Department

Department updated successfully!

EMPLOYEE MANAGEMENT SYSTEM

Department

Department Management

Choose action

Delete Department

Enter Department ID to Delete

8

Delete Department

Department deleted successfully!

- Next page created to add, update and delete attendance details.

```

elif(choice=="Attendance"):
    st.subheader("Attendance Management")
    att_action = st.selectbox("Choose action", ("None", "Mark Attendance", "Update Attendance", "Delete Attendance"))

    if att_action == "Mark Attendance":
        attendance_id = st.text_input("Enter Attendance ID")
        eid = st.text_input("Enter Employee ID")
        date = st.date_input("Select Date")
        check_in = st.time_input("Check-In Time")
        check_out = st.time_input("Check-Out Time")
        status = st.selectbox("Select Status", ["Present", "Absent", "Leave"])

        if st.button("Add Attendance"):
            mydb = mysql.connector.connect(host="localhost", user="root", password="1234", database="company")
            c = mydb.cursor()
            c.execute("INSERT INTO attendance (attendance_id, employee_id, date, check_in, check_out, status) VALUES (%s, %s, %s, %s, %s, %s)", (attendance_id, eid, date, check_in, check_out, status))
            mydb.commit()
            c.close()
            mydb.close()
            st.success("Attendance marked successfully!")

    elif att_action == "Update Attendance":
        attendance_id = st.text_input("Enter Attendance ID")
        update_field = st.selectbox("Select Field to Update", ["attendance_id", "employee_id", "date", "check_in", "check_out", "status"])
        new_value = st.text_input("Enter New Value")
        update_btn = st.button("Update Attendance")

        if update_btn:
            mydb = mysql.connector.connect(host="localhost", user="root", password="1234", database="company")
            c = mydb.cursor()
            c.execute(f"UPDATE attendance SET {update_field} = %s WHERE attendance_id = %s", (new_value, attendance_id))
            mydb.commit()
            c.close()
            mydb.close()
            st.success("Attendance updated successfully !")

    elif att_action == "Delete Attendance":
        attendance_id = st.text_input("Enter attendance ID to Delete")
        delete_att_btn = st.button("Delete Attendance")

        if delete_att_btn:
            mydb = mysql.connector.connect(host="localhost", user="root", password="1234", database="company")
            c = mydb.cursor()
            c.execute("DELETE FROM attendance WHERE attendance_id = %s", (attendance_id,))
            mydb.commit()
            c.close()
            mydb.close()
            st.success("Attendance deleted successfully!")

```

Enter Employee ID
2

Select Date
2025/02/12

Check-In Time
09:00

Check-Out Time
06:00

Select Status
Present

Add Attendance

Attendance marked successfully!

Attendance

Attendance Management

Choose action

Update Attendance

Enter Attendance ID

4

Select Field to Update

check_in

Enter New Value

8

Update Attendance

Attendance updated successfully !



My Menu

Attendance

EMPLOYEE MANAGEMENT SYSTEM

Attendance Management

Choose action

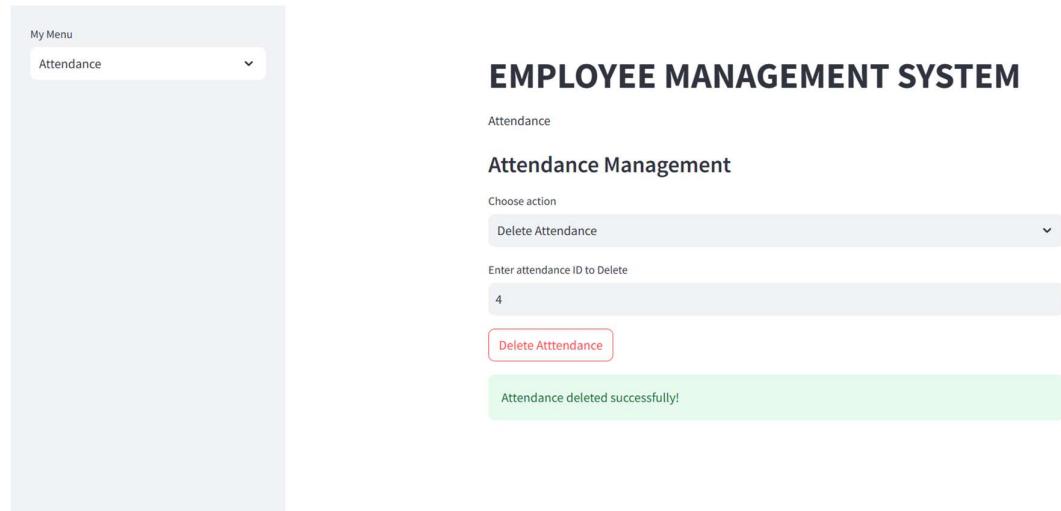
Delete Attendance

Enter attendance ID to Delete

4

Delete Attendance

Attendance deleted successfully!



- Finally create a session to add, update and delete performance details.

```

elif choice == "Performance":
    st.subheader("Performance Management")
    perf_action = st.selectbox("Choose action", ("None", "Add Performance", "Update Performance", "Delete Performance"))

if perf_action == "None":
    st.video("https://www.youtube.com/watch?v=fokvKvhIx8")

elif perf_action == "Add Performance":
    perf_id = st.text_input("Enter Performance ID")
    eid = st.text_input("Enter Employee ID")
    eval_date = st.date_input("Evaluation Date", value=date.today()) # Default to today's date
    rating = st.slider("Performance Rating (1 to 5)", 1.0, 5.0, step=0.1)
    review = st.text_area("Performance Review", key="review")
    add_perf_btn = st.button("Submit Performance Review")

    if add_perf_btn:
        mydb = mysql.connector.connect(host="localhost", user="root", password="1234", database="company")
        c = mydb.cursor()
        c.execute("INSERT INTO performance (performance_id, employee_id, evaluation_date, rating, feedback) VALUES (%s,%s, %s, %s,%s)", (perf_id,eid,eval_date,rating, review))
        mydb.commit()
        st.success("Performance review submitted successfully!")

elif perf_action == "Update Performance":
    perf_id = st.text_input("Enter Performance ID")
    update_field = st.selectbox("Select Field to Update", ["perf_id", "emp_id", "rating", "review"])
    new_value = st.text_input("Enter New Value")
    update_btn = st.button("Update Performance")

    if update_btn:
        mydb = mysql.connector.connect(host="localhost", user="root", password="1234", database="company")
        c = mydb.cursor()
        c.execute(f"UPDATE performance SET {update_field} = %s WHERE performance_id = %s", (new_value, perf_id))
        mydb.commit()
        c.close()
        mydb.close()
        st.success("Performance updated successfully !")

elif perf_action == "Delete Performance":
    perf_id = st.text_input("Enter Performance ID to Delete")
    delete_perf_btn = st.button("Delete Performance")

    if delete_perf_btn:
        mydb = mysql.connector.connect(host="localhost", user="root", password="1234", database="company")
        c = mydb.cursor()
        c.execute("DELETE FROM performance WHERE performance_id = %s", (perf_id,))
        mydb.commit()
        c.close()
        mydb.close()
        st.success("Performance deleted successfully!")

```

- In this performance page I create one performance related video.
- Also provided space to make rating and feedback.

EMPLOYEE MANAGEMENT SYSTEM

Performance

Performance Management

Choose action

None



EMPLOYEE MANAGEMENT SYSTEM

Performance

Performance Management

Choose action

Add Performance

Enter Performance ID

3

Enter Employee ID

3

Evaluation Date

2025/02/19

Performance Rating (1 to 5)



Performance Review

outstanding

Submit Performance Review

Performance review submitted successfully!

EMPLOYEE MANAGEMENT SYSTEM

Performance

Performance Management

Choose action

Update Performance

Enter Performance ID

3

Select Field to Update

rating

Enter New Value

4.5

Update Performance

Performance updated successfully !

EMPLOYEE MANAGEMENT SYSTEM

Performance

Performance Management

Choose action

Delete Performance

Enter Performance ID to Delete

3

Delete Performance

Performance deleted successfully!

- These are the features I created in this project.

Conclusion

The Employee Management System (EMS) developed in this project provides a comprehensive solution for managing employee-related tasks efficiently. With features such as user registration, login, and logout, the system ensures secure access for authorized personnel. The ability to manage employee details, salary, department, attendance, and performance through add, update, and delete functionalities makes it a robust and user-friendly application.

The ability to view structured data in tables enhances transparency and allows for better decision-making. Additionally, the system improves record-keeping and ensures that employee information is managed systematically and securely.

Hence this Employee Management System offers an efficient, scalable, and reliable solution for workforce management.