**Code:**  
  
import mysql.connector

from mysql.connector import Error

import hashlib

from dotenv import load\_dotenv

import os

load\_dotenv()

database\_pwd = os.getenv('DATABASE\_PWD')

database\_user=os.getenv('DATABASE\_USER')

admin\_username=os.getenv('ADMIN\_USERNAME')

admin\_password=os.getenv('ADMIN\_PASSWORD')

def create\_connection():

    try:

        connection = mysql.connector.connect(

            host='localhost',

            user=database\_user,

            password=database\_pwd,

            database='employee\_management'

        )

        if connection.is\_connected():

            return connection

    except Error as e:

        print(f"Error: {e}")

        return None

def hash\_password(password):

    return hashlib.sha256(password.encode()).hexdigest()

def check\_employee\_login(cursor, email, password):

    hashed\_password = hash\_password(password)

    query = "SELECT \* FROM employees WHERE email = %s AND password = %s"

    cursor.execute(query, (email, hashed\_password))

    return cursor.fetchone()

def get\_employee\_by\_id(cursor, emp\_id):

    query = "SELECT \* FROM employees WHERE id = %s"

    cursor.execute(query, (emp\_id,))

    return cursor.fetchone()

def get\_all\_employees(cursor):

    query = "SELECT \* FROM employees"

    cursor.execute(query)

    return cursor.fetchall()

def modify\_employee(cursor, emp\_id, name=None, salary=None, dept=None, password=None):

    update\_fields = []

    params = []

    if name:

        update\_fields.append("name = %s")

        params.append(name)

    if salary:

        update\_fields.append("salary = %s")

        params.append(salary)

    if dept:

        update\_fields.append("dept = %s")

        params.append(dept)

    if password:

        update\_fields.append("password = %s")

        params.append(hash\_password(password))

    params.append(emp\_id)

    if update\_fields:

        query = f"UPDATE employees SET {', '.join(update\_fields)} WHERE id = %s"

        cursor.execute(query, tuple(params))

        connection.commit()  # Commit the changes

def add\_employee(cursor, email, name, password, salary, dept):

    hashed\_password = hash\_password(password)

    query = "INSERT INTO employees (email, name, password, salary, dept) VALUES (%s, %s, %s, %s, %s)"

    cursor.execute(query, (email, name, hashed\_password, salary, dept))

    connection.commit()  # Commit the changes

def delete\_employee(cursor, emp\_id):

    query = "DELETE FROM employees WHERE id = %s"

    cursor.execute(query, (emp\_id,))

    connection.commit()  # Commit the changes

def admin\_operations(cursor):

    while True:

        print("\nAdmin Menu:")

        print("1. Add Employee")

        print("2. View Employee Details")

        print("3. View All Employees")

        print("4. Delete Employee")

        print("5. Modify Employee")

        print("6. Exit")

        choice = input("Enter your choice: ")

        if choice == '1':

            email = input("Enter email: ")

            name = input("Enter name: ")

            password = input("Enter password: ")

            salary = float(input("Enter salary: "))

            dept = input("Enter department: ")

            add\_employee(cursor, email, name, password, salary, dept)

            print("Employee added successfully!")

        elif choice == '2':

            emp\_id = int(input("Enter employee id: "))

            employee = get\_employee\_by\_id(cursor, emp\_id)

            if employee:

                print(f"Employee Details: {employee}")

            else:

                print("Employee not found.")

        elif choice == '3':

            employees = get\_all\_employees(cursor)

            if employees:

                print("All Employees:")

                for emp in employees:

                    print(emp)

            else:

                print("No employees found.")

        elif choice == '4':

            emp\_id = int(input("Enter employee id to delete: "))

            employee = get\_employee\_by\_id(cursor, emp\_id)

            if employee:

                delete\_employee(cursor, emp\_id)

                print("Employee deleted successfully!")

            else:

                print("Employee not found. Cannot delete.")

        elif choice == '5':

            emp\_id = int(input("Enter employee id to modify: "))

            employee = get\_employee\_by\_id(cursor, emp\_id)

            if employee:

                name = input("Enter new name (press Enter to skip): ")

                salary = input("Enter new salary (press Enter to skip): ")

                dept = input("Enter new department (press Enter to skip): ")

                password = input("Enter new password (press Enter to skip): ")

                modify\_employee(cursor, emp\_id, name, salary, dept, password)

                print("Employee details updated successfully!")

            else:

                print("Employee not found. Cannot modify.")

        elif choice == '6':

            break

        else:

            print("Invalid choice. Please try again.")

def employee\_operations(cursor, emp\_id):

    while True:

        print("\nEmployee Menu:")

        print("1. View Details")

        print("2. Modify Details")

        print("3. Exit")

        choice = input("Enter your choice: ")

        if choice == '1':

            employee = get\_employee\_by\_id(cursor, emp\_id)

            if employee:

                print(f"Employee Details: {employee}")

            else:

                print("Employee not found.")

        elif choice == '2':

            name = input("Enter new name (press Enter to skip): ")

            salary = input("Enter new salary (press Enter to skip): ")

            dept = input("Enter new department (press Enter to skip): ")

            password = input("Enter new password (press Enter to skip): ")

            modify\_employee(cursor, emp\_id, name, salary, dept, password)

            print("Employee details updated successfully!")

        elif choice == '3':

            break

        else:

            print("Invalid choice. Please try again.")

def main():

    global connection

    connection = create\_connection()

    if connection:

        cursor = connection.cursor()

        print()

        print("                                                  Employee Management System")

        print()

        input("Press Enter to Continue")

        print()

        user\_type = input("Are you an 'admin' or 'employee'? ").strip().lower()

        if user\_type == 'admin':

            admin\_username = input("Enter admin username: ")

            admin\_password = input("Enter admin password: ")

            # Assuming admin credentials validation

            if admin\_username ==admin\_username  and admin\_password ==admin\_password :

                admin\_operations(cursor)

            else:

                print("Invalid admin credentials.")

        elif user\_type == 'employee':

            email = input("Enter your email: ")

            password = input("Enter your password: ")

            employee = check\_employee\_login(cursor, email, password)

            if employee:

                emp\_id = employee[0]

                print("Employee login successful!")

                employee\_operations(cursor, emp\_id)

            else:

                print("Invalid employee credentials.")

        else:

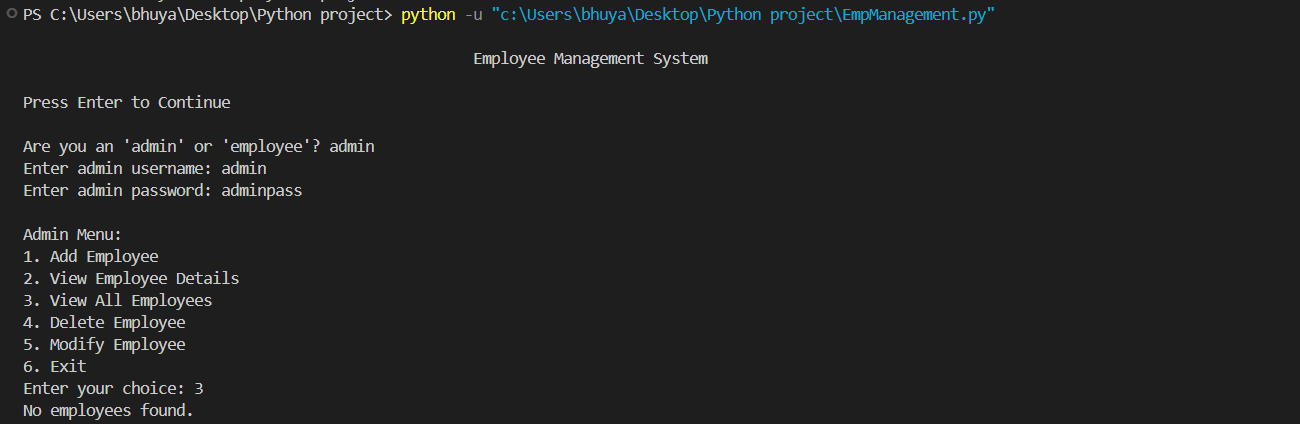
            print("Invalid user type.")

        cursor.close()

        connection.close()

if \_\_name\_\_ == "\_\_main\_\_":

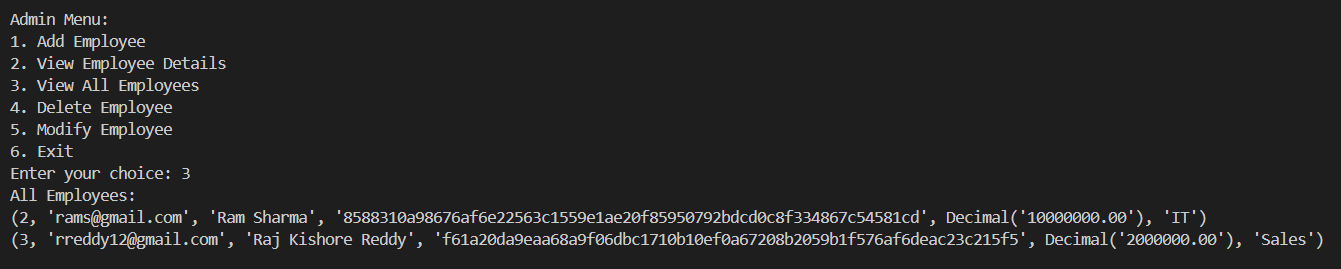
    main()

**Executions**  
  
**Login by Admin and viewing all employees**  
  


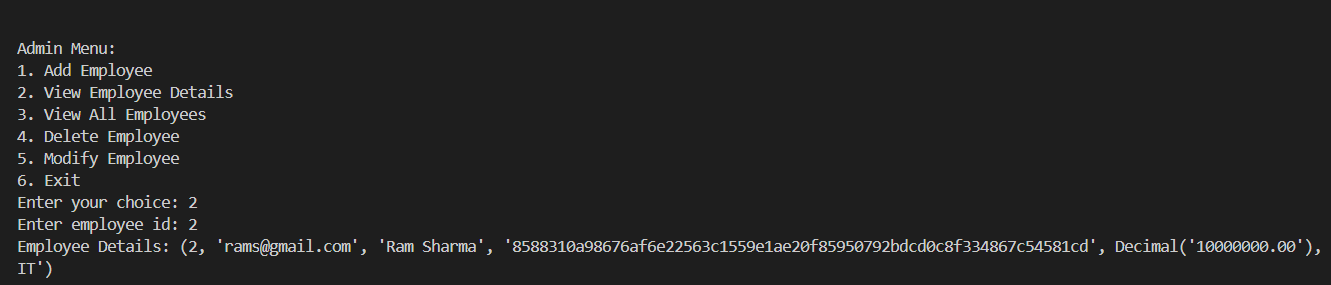
**Adding new employee**



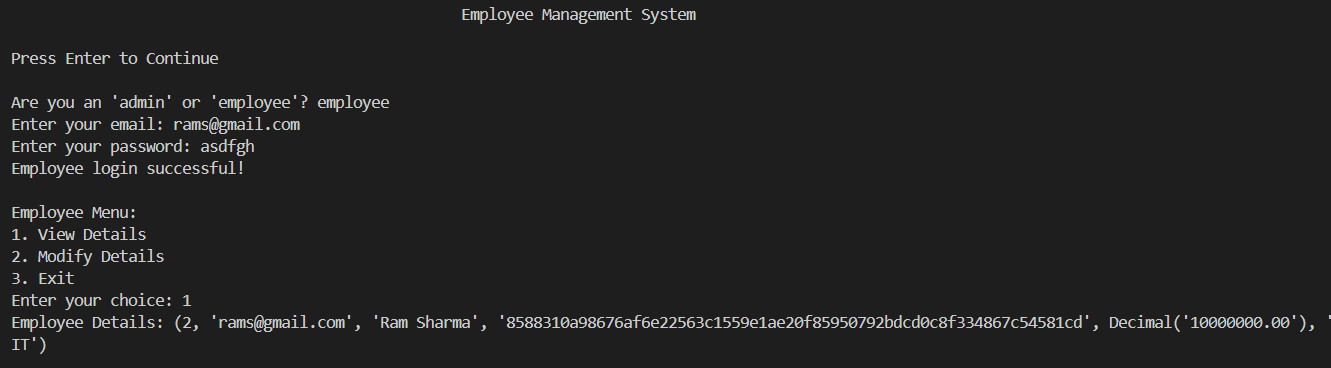
**Viewing all Employee Details**



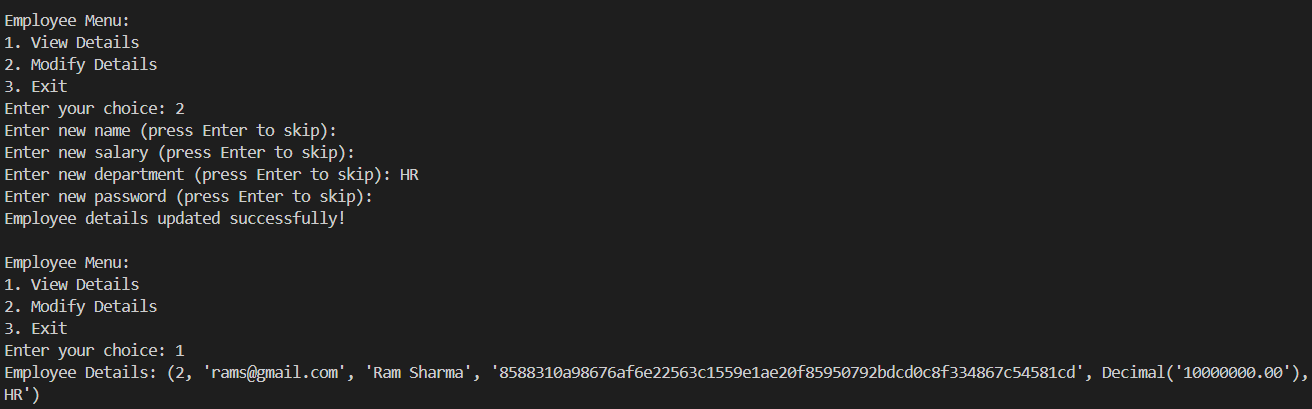
**Viewing a single Employee by ID**



**Employee Login and viewing of details**



**Employee Modifying their data**



**Deletion of an Employee by the Admin**

