Aakriti Singh Dikhit, 206704

Employee Management System using Python Project Report

1. Introduction

The Employee Management System is a console-based application designed to manage employee data efficiently. The system allows users to perform various operations, including inserting new employee records, viewing all employees or specific employees by ID, updating existing employee details, and deleting employee records. This system utilizes a MySQL database to store and manage employee data, providing a user-friendly interface for interacting with the database through a main menu-driven approach.

Key Features:

- **Insert Employee**: Add new employee records into the database.
- **View Employees**: Display a list of all employees.
- **View Employee by ID**: Retrieve and display details of a specific employee by their ID.
- **Update Employee**: Modify details of an existing employee.
- **Delete Employee**: Remove an employee record from the database

2. Project Code

2.1 Main Menu (main_menu.py)

```
import mysql.connector
import InsertData
import ViewAll
import ViewbyId
import UpdateEmp
import DeleteEmp

def get_db_password():
    return input("Enter MySQL database password: ")

def connect_to_database(password):
    try:
        return mysql.connector.connect(
            host="localhost",
            user="root",
            passwd=password,
            database="amdocsproject"
        )
    except mysql.connector.Error as err:
        print(f"Error: {err}")
        return None
def show_menu():
    print("\n--- Employee Management System ---")
    print("1. Insert Employee")
    print("2. View Employees")
    print("3. View Employee by ID")
    print("4. Update Employee")
    print("5. Delete Employee")
    print("6. Exit")

def main():
    password = get_db_password()
    db = connect_to_database(password)

    if db is None:
        print("Unable to connect to the database. Exiting...")
        return

    while True:
        try:
            show_menu()
            choice = input("Enter your choice: ")

            if choice == '1':
                InsertData.insert_employee()
            elif choice == '2':
                ViewAll.view_employee()
            elif choice == '3':
                emp_id = input("Enter employee ID: ")
                ViewbyId.view_employee_by_id(emp_id)
            elif choice == '4':
                UpdateEmp.update_employee()
            elif choice == '5':
                DeleteEmp.delete_employee()
            elif choice == '6':
                print("Exiting...")
                break
            else:
                print("Invalid choice. Please try again.")
        except Exception as e:
            print(f"An error occurred: {e}")

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

2.2 Insert Employee (InsertData.py)

```
import mysql.connector

def insert_employee():
    database = mysql.connector.connect(
        host="localhost",
        user="root",
        passwd="18082002",
        database="amdocsproject"
    )

    cursorObject = database.cursor()
    emp_id = int(input("Enter Id: "))
    name = input("Enter Name: ")
    dept = input("Enter Department: ")
    age = int(input("Enter Age: "))
    email = input("Enter Email: ")

    sql = "Insert into employee(Employee_Id, Name, Department, Age, Email)" \
          "Values (%s,%s,%s,%s,%s)"
    val=(emp_id,name,dept,age,email)

    cursorObject.execute(sql, val)
    database.commit()
    database.close()
```

2.3 View All Employees (ViewAll.py)

```
import mysql.connector
def connect_to_database():
    return mysql.connector.connect(
        host="localhost",
        user="root",
        passwd="18082002",
        database="amdocsproject"
    )

def view_employee():
    db = connect_to_database()
    if db is None:
        return
    cursor = db.cursor()
    cursor.execute("SELECT * FROM employee")
    results = cursor.fetchall()
    if results:
        print("\nEmployee Records:")
        for row in results:
            print(f"ID: {row[0]}, Name: {row[1]}, Department: {row[2]}, Salary: 
{row[3]}, Email: {row[4]}")
    else:
        print("No employee records found.")
    cursor.close()
    db.close()

if __name__ == "__main__":
    view_employee()
```

2.4 View Employee by ID (ViewbyId.py)

```
import mysql.connector
def connect_to_database():
    return mysql.connector.connect(
        host="localhost",
        user="root",
        passwd="18082002",
        database="amdocsproject"
    )
def view_employee_by_id(emp_id):
    #emp_id = int(input("Enter employee ID to view: "))
    db = connect_to_database()
    if db is None:
        return
    cursor = db.cursor()
    cursor.execute("SELECT * FROM employee WHERE Employee_Id = %s", 
(emp_id,))
    result = cursor.fetchone()
    if result:
        print(f"ID: {result[0]}, Name: {result[1]}, Department: {result[2]}, Salary: 
{result[3]}, Email: {result[4]}")
    else:
        print("No employee found with that ID.")
    cursor.close()
    db.close()
if __name__ == "__main__":
    view_employee_by_id()
```

2.5 Update Employee (UpdateEmp.py)

```
import mysql.connector
def connect_to_database():
    return mysql.connector.connect(
        host="localhost",
        user="root",
        passwd="18082002",
        database="amdocsproject"
    )

def update_employee():
    Employee_Id = int(input("Enter employee ID to update: "))
    db = connect_to_database()
    if db is None:
        return
    cursor = db.cursor()

    cursor.execute("SELECT * FROM employee WHERE Employee_Id = %s", 
(Employee_Id,))
    result = cursor.fetchone()

    if result:
        print(f"Current details: ID: {result[0]}, Name: {result[1]}, Department: 
{result[2]}, Age: {result[3]}, Email: {result[4]}")

        Name = input("Enter new name (or leave blank to keep current): ") or 
result[1]
        Department = input("Enter new department (or leave blank to keep current): 
") or result[2]
        Age = input("Enter new age (or leave blank to keep current): ") or result[3]
        Email = input("Enter new email (or leave blank to keep current): ") or
result[4]

        sql = """UPDATE employee SET Name = %s, Department = %s, Age = %s,
Email = %s WHERE Employee_Id = %s"""
        cursor.execute(sql, (Name, Department, Age, Email, Employee_Id))

        db.commit()
        print("Employee details updated successfully.")
    else:
        print("No employee found with that ID.")

    cursor.close()
    db.close()

if __name__ == "__main__":
    update_employee()
```

2.6 Delete Employee (DeleteEmp.py)

```
import mysql.connector
def connect_to_database():
    return mysql.connector.connect(
        host="localhost",
        user="root",
        passwd="18082002",
        database="amdocsproject"
    )

def delete_employee():
    emp_id = int(input("Enter employee ID to delete: "))
    db = connect_to_database()
    if db is None:
        return
    cursor = db.cursor()
    cursor.execute("SELECT * FROM employee WHERE Employee_Id = %s",
(emp_id,))
    result = cursor.fetchone()
    if result:
        cursor.execute("DELETE FROM employee WHERE Employee_Id = %s",
(emp_id,))
```

```
        db.commit()
        print("Employee deleted successfully.")
    else:
        print("No employee found with that ID.")
    cursor.close()
    db.close()


if __name__ == "__main__":
    delete_employee()
```

3. Supporting Files:

3.1 Database Connection (ConnectDB.py)

```
import mysql.connector

database = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="18082002"
)

cursorObject = database.cursor()
cursorObject.execute("create database AmdocsProject")
```

3.2 Table Creation (CreateTable.py)

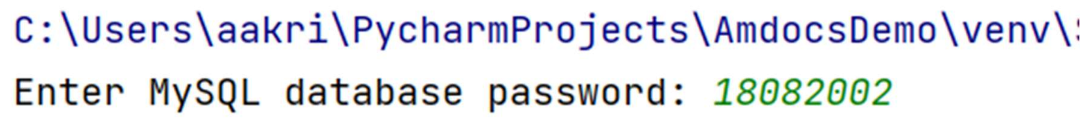
```
import mysql.connector

database = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="18082002",
    database="AmdocsProject"
)

cursorObject = database.cursor()

studentRecord = """Create Table Employee(
    Employee_Id int not null,
    Name Varchar(50) not null,
    Deptartment Varchar(50),
    Age int,
    Email varchar(50))
    """
cursorObject.execute(studentRecord)
database.close()
```

3.2 Inserting Multiple Data (InsertData2.py)

```
import mysql.connector
def connect_to_database():
    return mysql.connector.connect(
        host="localhost",
        user="root",
        passwd="18082002",
        database="amdocsproject"
    )

def insert_default_data():
    db = connect_to_database()
    cursor = db.cursor()

    employee = [
        (3,"Ravi Kumar", "HR",24, "ravi.kumar@example.com"),
        (4,"Priya Sharma", "Finance", 23, "priya.sharma@example.com"),
        (5,"Amit Verma", "IT", 29, "amit.verma@example.com"),
        (6,"Sneha Patel", "Marketing", 22, "sneha.patel@example.com"),
        (7,"Rajesh Singh", "Sales", 27, "rajesh.singh@example.com")
    ]

    sql = "INSERT INTO employee (Employee_Id, Name, Department, Age, email)
VALUES (%s,%s, %s, %s, %s)"
    cursor.executemany(sql, employee)
    db.commit()
    print(f"{cursor.rowcount} employees inserted successfully.")

    cursor.close()
    db.close()

if __name__ == "__main__":
    insert_default_data()
```

4. Output:

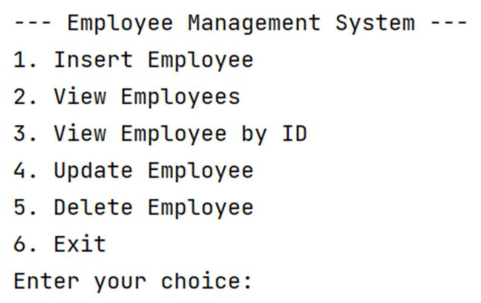
→Authentication for Database Connection



```
C:\Users\aakri\PycharmProjects\AmdocsDemo\venv\
Enter MySQL database password: 18082002
```

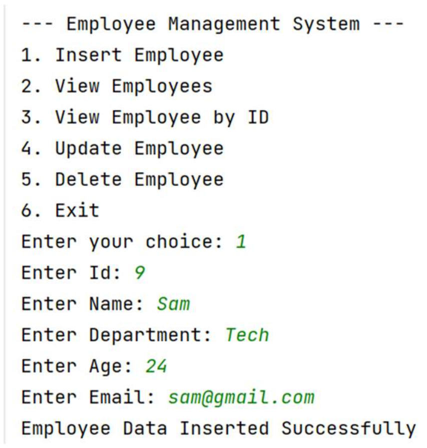
--

→Main Menu Show

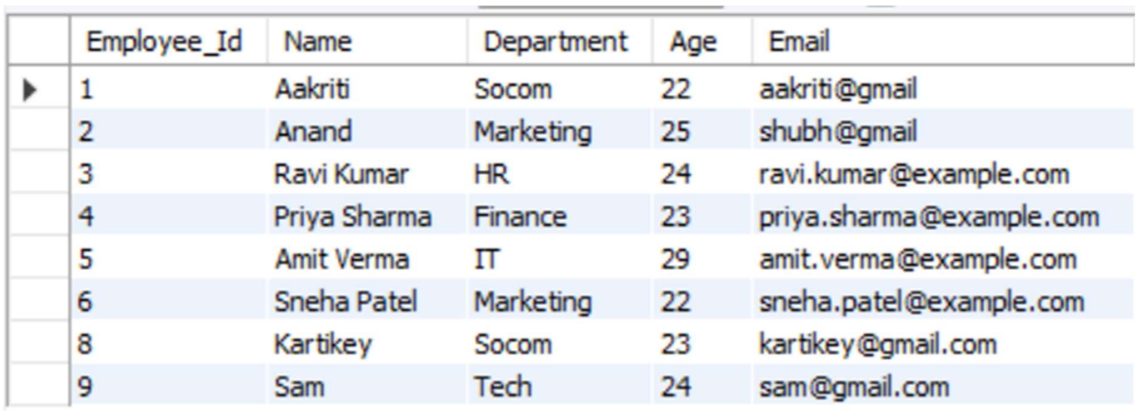


```
--- Employee Management System ---
1. Insert Employee
2. View Employees
3. View Employee by ID
4. Update Employee
5. Delete Employee
6. Exit
Enter your choice:
```

4.1. Insert Employee

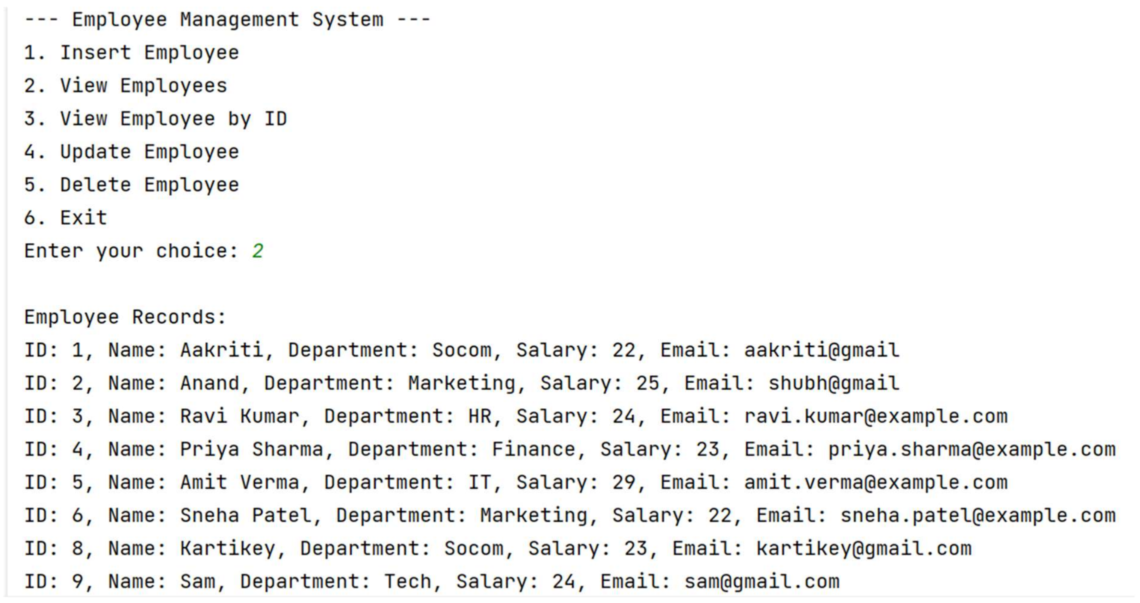


```
--- Employee Management System ---
1. Insert Employee
2. View Employees
3. View Employee by ID
4. Update Employee
5. Delete Employee
6. Exit
Enter your choice: 1
Enter Id: 9
Enter Name: Sam
Enter Department: Tech
Enter Age: 24
Enter Email: sam@gmail.com
Employee Data Inserted Successfully
```



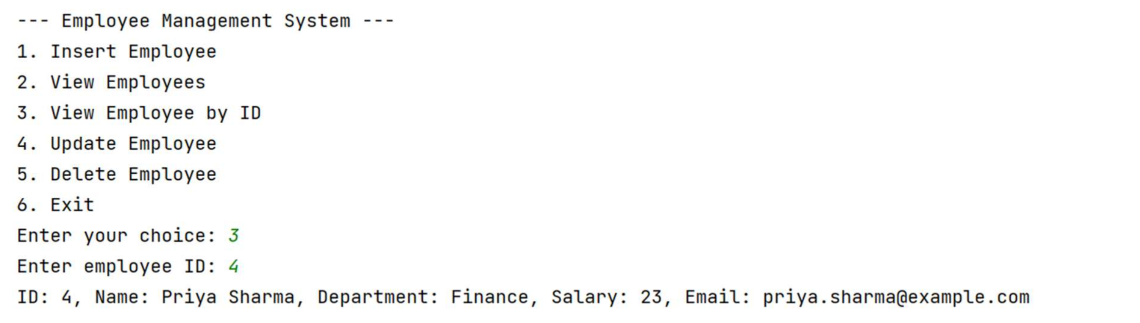
	Employee_Id	Name	Department	Age	Email
▶	1	Aakriti	Socom	22	aakriti@gmail
	2	Anand	Marketing	25	shubh@gmail
	3	Ravi Kumar	HR	24	ravi.kumar@example.com
	4	Priya Sharma	Finance	23	priya.sharma@example.com
	5	Amit Verma	IT	29	amit.verma@example.com
	6	Sneha Patel	Marketing	22	sneha.patel@example.com
	8	Kartikey	Socom	23	kartikey@gmail.com
	9	Sam	Tech	24	sam@gmail.com

4.2. View All Employees



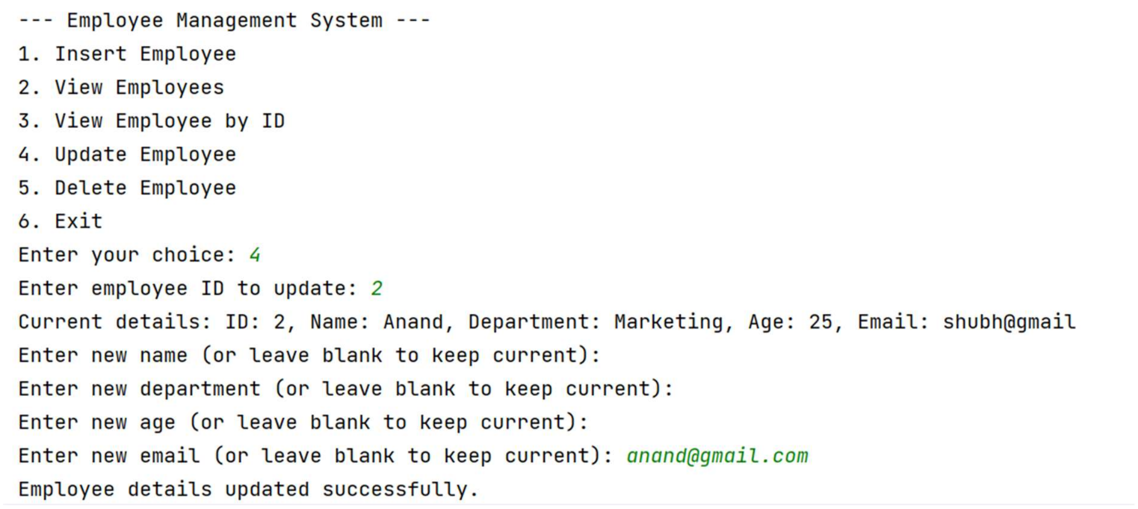
```
--- Employee Management System ---
1. Insert Employee
2. View Employees
3. View Employee by ID
4. Update Employee
5. Delete Employee
6. Exit
Enter your choice: 2

Employee Records:
ID: 1, Name: Aakriti, Department: Socom, Salary: 22, Email: aakriti@gmail
ID: 2, Name: Anand, Department: Marketing, Salary: 25, Email: shubh@gmail
ID: 3, Name: Ravi Kumar, Department: HR, Salary: 24, Email: ravi.kumar@example.com
ID: 4, Name: Priya Sharma, Department: Finance, Salary: 23, Email: priya.sharma@example.com
ID: 5, Name: Amit Verma, Department: IT, Salary: 29, Email: amit.verma@example.com
ID: 6, Name: Sneha Patel, Department: Marketing, Salary: 22, Email: sneha.patel@example.com
ID: 8, Name: Kartikey, Department: Socom, Salary: 23, Email: kartikey@gmail.com
ID: 9, Name: Sam, Department: Tech, Salary: 24, Email: sam@gmail.com
```

4.3. View Employee by ID

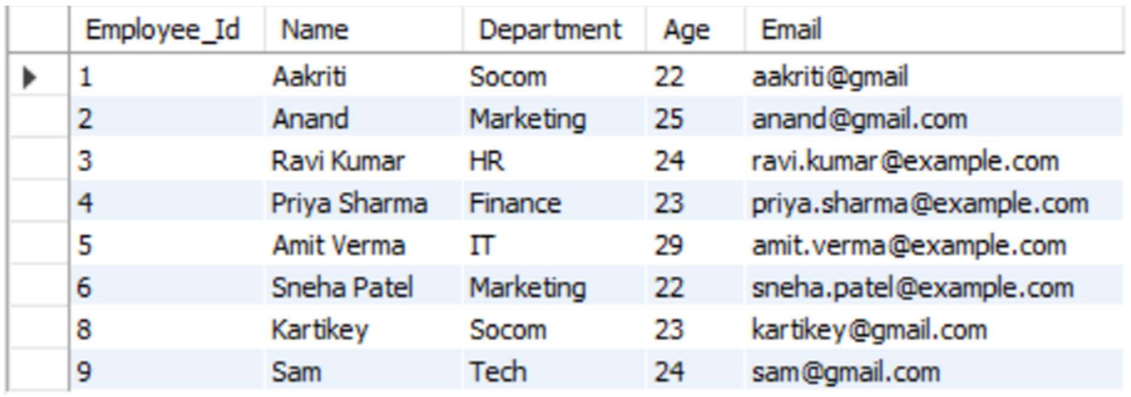


```
--- Employee Management System ---
1. Insert Employee
2. View Employees
3. View Employee by ID
4. Update Employee
5. Delete Employee
6. Exit
Enter your choice: 3
Enter employee ID: 4
ID: 4, Name: Priya Sharma, Department: Finance, Salary: 23, Email: priya.sharma@example.com
```

4.4. Update Employee



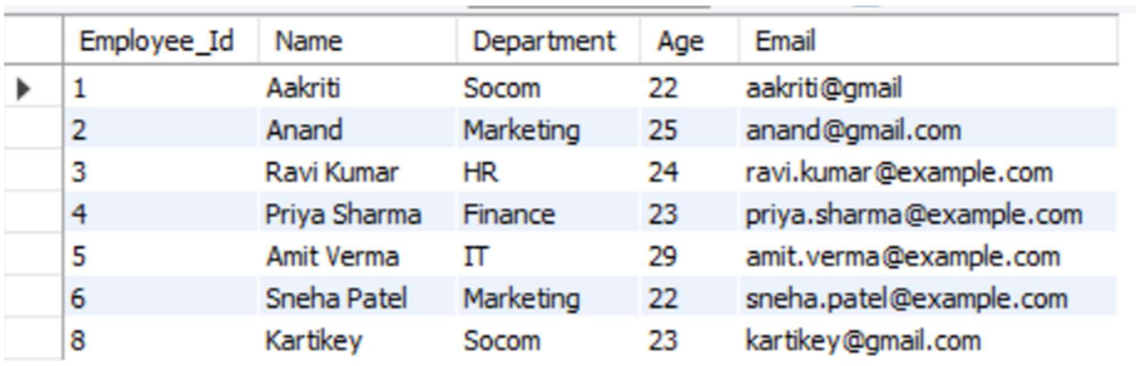
```
--- Employee Management System ---
1. Insert Employee
2. View Employees
3. View Employee by ID
4. Update Employee
5. Delete Employee
6. Exit
Enter your choice: 4
Enter employee ID to update: 2
Current details: ID: 2, Name: Anand, Department: Marketing, Age: 25, Email: shubh@gmail
Enter new name (or leave blank to keep current):
Enter new department (or leave blank to keep current):
Enter new age (or leave blank to keep current):
Enter new email (or leave blank to keep current): anand@gmail.com
Employee details updated successfully.
```



	Employee_Id	Name	Department	Age	Email
▶	1	Aakriti	Socom	22	aakriti@gmail
	2	Anand	Marketing	25	anand@gmail.com
	3	Ravi Kumar	HR	24	ravi.kumar@example.com
	4	Priya Sharma	Finance	23	priya.sharma@example.com
	5	Amit Verma	IT	29	amit.verma@example.com
	6	Sneha Patel	Marketing	22	sneha.patel@example.com
	8	Kartikey	Socom	23	kartikey@gmail.com
	9	Sam	Tech	24	sam@gmail.com

4.5. Delete Employee

```
--- Employee Management System ---
1. Insert Employee
2. View Employees
3. View Employee by ID
4. Update Employee
5. Delete Employee
6. Exit
Enter your choice: 5
Enter employee ID to delete: 9
Employee deleted successfully.
```



Employee_Id	Name	Department	Age	Email
1	Aakriti	Socom	22	aakriti@gmail
2	Anand	Marketing	25	anand@gmail.com
3	Ravi Kumar	HR	24	ravi.kumar@example.com
4	Priya Sharma	Finance	23	priya.sharma@example.com
5	Amit Verma	IT	29	amit.verma@example.com
6	Sneha Patel	Marketing	22	sneha.patel@example.com
8	Kartikey	Socom	23	kartikey@gmail.com

4.6 Exit

```
--- Employee Management System ---
1. Insert Employee
2. View Employees
3. View Employee by ID
4. Update Employee
5. Delete Employee
6. Exit
Enter your choice: 6
Exiting...
```

----

5. Conclusion

The Employee Management System project successfully demonstrates the integration of a Python-based application with a MySQL database to handle various employee-related operations. Through this project, we have developed a comprehensive console-based application that allows users to efficiently manage employee data with functionalities for inserting, viewing, updating, and deleting employee records.