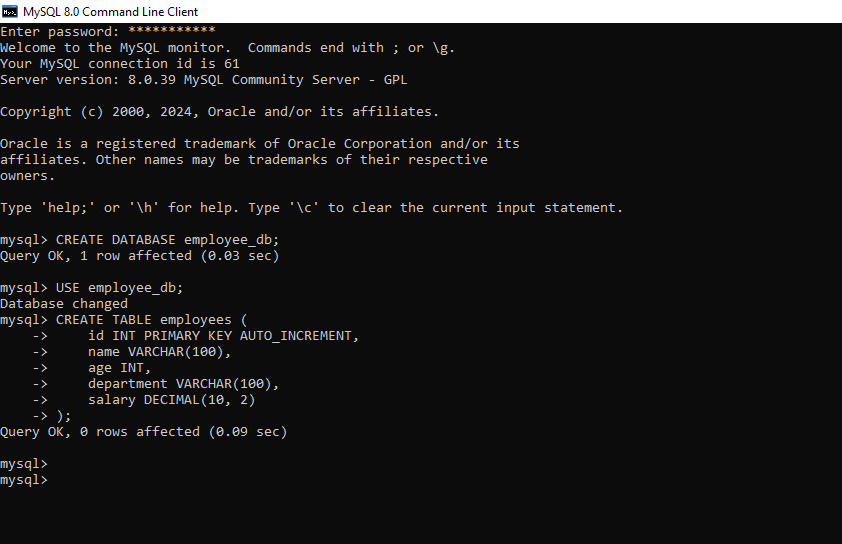
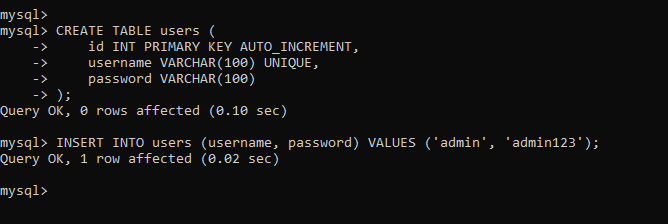
**EMPLOYEMENT MANAGEMENT SYSTEM**

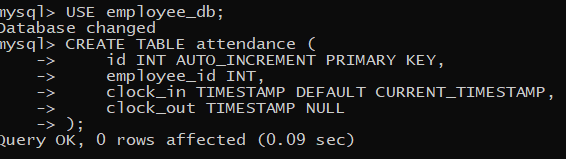
**Mansi Karadkhedkar (206607)**

The **Employee Management System** is a console-based application developed using **Python** and **MySQL** to streamline the management of employee records and attendance. The project focuses on performing basic **CRUD operations**—Create, Read, Update, Delete—while maintaining the integrity and security of employee data. Additionally, the system includes an **attendance management feature** that tracks employees’ work hours through clock-in and clock-out functionality.

This system is designed to be simple yet robust, enabling organizations to handle employee data efficiently. The project also incorporates a **login system** with access control, ensuring that only authorized users can interact with the system. The **attendance system** provides detailed logs of employee work hours, offering valuable insights into productivity.

DATABASES CODE:

****

****

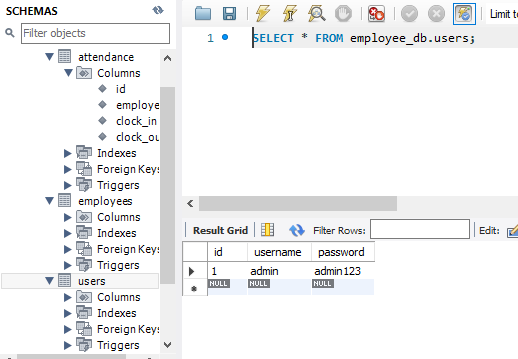
**USERS.PY:**

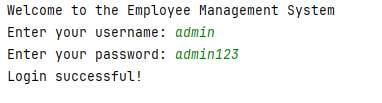
import mysql.connector  
  
*# Establish connection to MySQL*def connect\_to\_db():  
 return mysql.connector.connect(  
 host="localhost",  
 user="root", *# Replace with your MySQL username* password="Redminote#9", *# Replace with your MySQL password* database="employee\_db"  
 )  
  
*# Login function to verify user credentials*def login():  
 db = connect\_to\_db()  
 cursor = db.cursor()  
  
 username = input("Enter your username: ")  
 password = input("Enter your password: ")  
  
 query = "SELECT \* FROM users WHERE username = %s AND password = %s"  
 cursor.execute(query, (username, password))  
 result = cursor.fetchone()  
  
 db.close()  
  
 if result:  
 print("Login successful!")  
 return True  
 else:  
 print("Invalid username or password. Try again.")  
 return False  
  
*# Create new employee record*def add\_employee():  
 db = connect\_to\_db()  
 cursor = db.cursor()  
  
 name = input("Enter employee name: ")  
 age = int(input("Enter employee age: "))  
 department = input("Enter department: ")  
 salary = float(input("Enter employee salary: "))  
  
 query = "INSERT INTO employees (name, age, department, salary) VALUES (%s, %s, %s, %s)"  
 cursor.execute(query, (name, age, department, salary))  
 db.commit()  
  
 print("Employee added successfully!")  
 db.close()  
  
*# Fetch and display all employee records*def view\_employees():  
 db = connect\_to\_db()  
 cursor = db.cursor()  
  
 cursor.execute("SELECT \* FROM employees")  
 records = cursor.fetchall()  
  
 print("ID | Name | Age | Department | Salary")  
 print("-------------------------------------")  
 for row in records:  
 print(f"{row[0]} | {row[1]} | {row[2]} | {row[3]} | {row[4]}")  
  
 db.close()  
  
*# Update employee record*def update\_employee():  
 db = connect\_to\_db()  
 cursor = db.cursor()  
  
 emp\_id = int(input("Enter employee ID to update: "))  
 new\_salary = float(input("Enter new salary: "))  
  
 query = "UPDATE employees SET salary = %s WHERE id = %s"  
 cursor.execute(query, (new\_salary, emp\_id))  
 db.commit()  
  
 print("Employee salary updated successfully!")  
 db.close()  
  
*# Delete employee record*def delete\_employee():  
 db = connect\_to\_db()  
 cursor = db.cursor()  
  
 emp\_id = int(input("Enter employee ID to delete: "))  
  
 query = "DELETE FROM employees WHERE id = %s"  
 cursor.execute(query, (emp\_id,))  
 db.commit()  
  
 print("Employee deleted successfully!")  
 db.close()  
  
  
*# ATTENDANCE  
# Employee Clock-In function*def clock\_in():  
 db = connect\_to\_db()  
 cursor = db.cursor()  
  
 emp\_id = int(input("Enter your Employee ID to clock in: "))  
  
 *# Check if the employee has already clocked in and not clocked out yet* query = "SELECT \* FROM attendance WHERE employee\_id = %s AND clock\_out IS NULL"  
 cursor.execute(query, (emp\_id,))  
 result = cursor.fetchone()  
  
 if result:  
 print(f"Employee {emp\_id} is already clocked in. Please clock out before clocking in again.")  
 else:  
 query = "INSERT INTO attendance (employee\_id) VALUES (%s)"  
 cursor.execute(query, (emp\_id,))  
 db.commit()  
 print(f"Employee {emp\_id} clocked in successfully!")  
  
 db.close()  
  
*# Employee Clock-Out function*def clock\_out():  
 db = connect\_to\_db()  
 cursor = db.cursor()  
  
 emp\_id = int(input("Enter your Employee ID to clock out: "))  
  
 *# Check if the employee has clocked in* query = "SELECT \* FROM attendance WHERE employee\_id = %s AND clock\_out IS NULL"  
 cursor.execute(query, (emp\_id,))  
 result = cursor.fetchone()  
  
 if result:  
 *# If the employee is clocked in, update the clock\_out time* query = "UPDATE attendance SET clock\_out = CURRENT\_TIMESTAMP WHERE employee\_id = %s AND clock\_out IS NULL"  
 cursor.execute(query, (emp\_id,))  
 db.commit()  
 print(f"Employee {emp\_id} clocked out successfully!")  
 else:  
 print(f"Employee {emp\_id} has not clocked in yet or has already clocked out.")  
  
 db.close()  
  
*# View attendance records for a specific employee*def view\_attendance():  
 db = connect\_to\_db()  
 cursor = db.cursor()  
  
 emp\_id = int(input("Enter Employee ID to view attendance: "))  
  
 query = "SELECT \* FROM attendance WHERE employee\_id = %s"  
 cursor.execute(query, (emp\_id,))  
 records = cursor.fetchall()  
  
 print("Attendance Records:")  
 print("ID | Employee ID | Clock In Time | Clock Out Time")  
 print("-----------------------------------------------")  
 for row in records:  
 print(f"{row[0]} | {row[1]} | {row[2]} | {row[3]}")  
  
 db.close()  
  
  
  
*# Menu-driven system*def menu():  
 while True:  
 print("\nEmployee Management System")  
 print("1. Add Employee")  
 print("2. View Employees")  
 print("3. Update Employee")  
 print("4. Delete Employee")  
 print("5. Clock In")  
 print("6. Clock Out")  
 print("7. View Attendance")  
 print("8. Exit")  
  
 choice = int(input("Enter your choice: "))  
  
 if choice == 1:  
 add\_employee()  
 elif choice == 2:  
 view\_employees()  
 elif choice == 3:  
 update\_employee()  
 elif choice == 4:  
 delete\_employee()  
 elif choice == 5:  
 clock\_in()  
 elif choice == 6:  
 clock\_out()  
 elif choice == 7:  
 view\_attendance()  
 elif choice == 8:  
 print("Exiting...")  
 break  
 else:  
 print("Invalid choice, please try again.")  
  
  
  
*# Main function that runs the login system first*def main():  
 print("Welcome to the Employee Management System")  
 login\_attempts = 3  
 while login\_attempts > 0:  
 if login():  
 menu()  
 break  
 else:  
 login\_attempts -= 1  
 print(f"Remaining attempts: {login\_attempts}")  
 else:  
 print("Too many failed login attempts. Exiting system.")  
  
*# Run the program*if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**ATTENDANCE.PY**

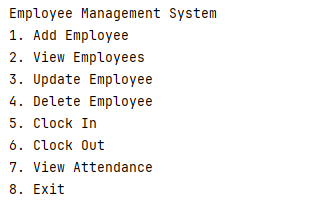
*# Employee Clock-In function*def clock\_in():  
 db = connect\_to\_db()  
 cursor = db.cursor()  
  
 emp\_id = int(input("Enter your Employee ID to clock in: "))  
  
 *# Check if the employee has already clocked in and not clocked out yet* query = "SELECT \* FROM attendance WHERE employee\_id = %s AND clock\_out IS NULL"  
 cursor.execute(query, (emp\_id,))  
 result = cursor.fetchone()  
  
 if result:  
 print(f"Employee {emp\_id} is already clocked in. Please clock out before clocking in again.")  
 else:  
 query = "INSERT INTO attendance (employee\_id) VALUES (%s)"  
 cursor.execute(query, (emp\_id,))  
 db.commit()  
 print(f"Employee {emp\_id} clocked in successfully!")  
  
 db.close()  
  
*# Employee Clock-Out function*def clock\_out():  
 db = connect\_to\_db()  
 cursor = db.cursor()  
  
 emp\_id = int(input("Enter your Employee ID to clock out: "))  
  
 *# Check if the employee has clocked in* query = "SELECT \* FROM attendance WHERE employee\_id = %s AND clock\_out IS NULL"  
 cursor.execute(query, (emp\_id,))  
 result = cursor.fetchone()  
  
 if result:  
 *# If the employee is clocked in, update the clock\_out time* query = "UPDATE attendance SET clock\_out = CURRENT\_TIMESTAMP WHERE employee\_id = %s AND clock\_out IS NULL"  
 cursor.execute(query, (emp\_id,))  
 db.commit()  
 print(f"Employee {emp\_id} clocked out successfully!")  
 else:  
 print(f"Employee {emp\_id} has not clocked in yet or has already clocked out.")  
  
 db.close()  
  
*# View attendance records for a specific employee*def view\_attendance():  
 db = connect\_to\_db()  
 cursor = db.cursor()  
  
 emp\_id = int(input("Enter Employee ID to view attendance: "))  
  
 query = "SELECT \* FROM attendance WHERE employee\_id = %s"  
 cursor.execute(query, (emp\_id,))  
 records = cursor.fetchall()  
  
 print("Attendance Records:")  
 print("ID | Employee ID | Clock In Time | Clock Out Time")  
 print("-----------------------------------------------")  
 for row in records:  
 print(f"{row[0]} | {row[1]} | {row[2]} | {row[3]}")  
  
 db.close()

1. **LOGIN**

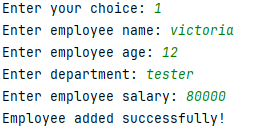
****

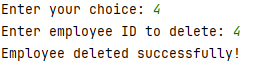
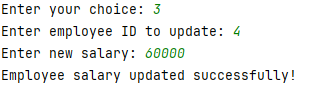
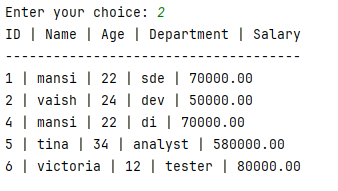
****

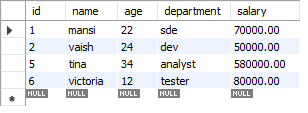
**2.FUNCTIONALITIES**

****

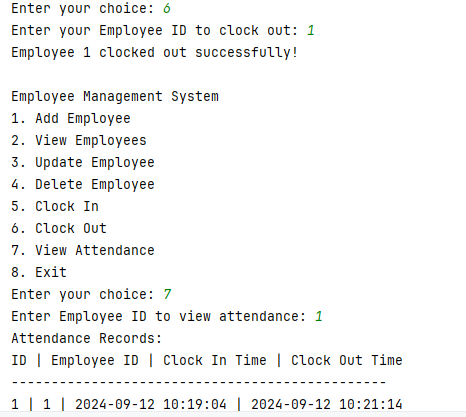
**3.OUTPUTS**

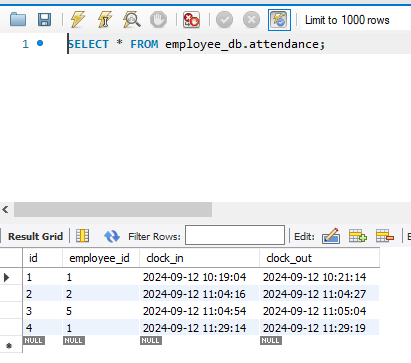
****

****

****

**4. ATTENDANCE**

****

****