Employee Management System

Name: Shilpi Kumari
Employee id : 206595

1. Intro

-> An employee management system that allows for employee registration, login, update, and removal.

->Streamline employee management tasks and provide an efficient way to handle employee data.

Project Features:

Employee Registration: New employees can register with details like name, location, joining date, salary, and password.

Employee Login: Allows employees to log in using their ID and password.

Update Details: Employees can update their own details (name, location, salary) after logging in.

Remove Employee: Admin (Shilpi) can remove employees after logging in.

Tech Stack

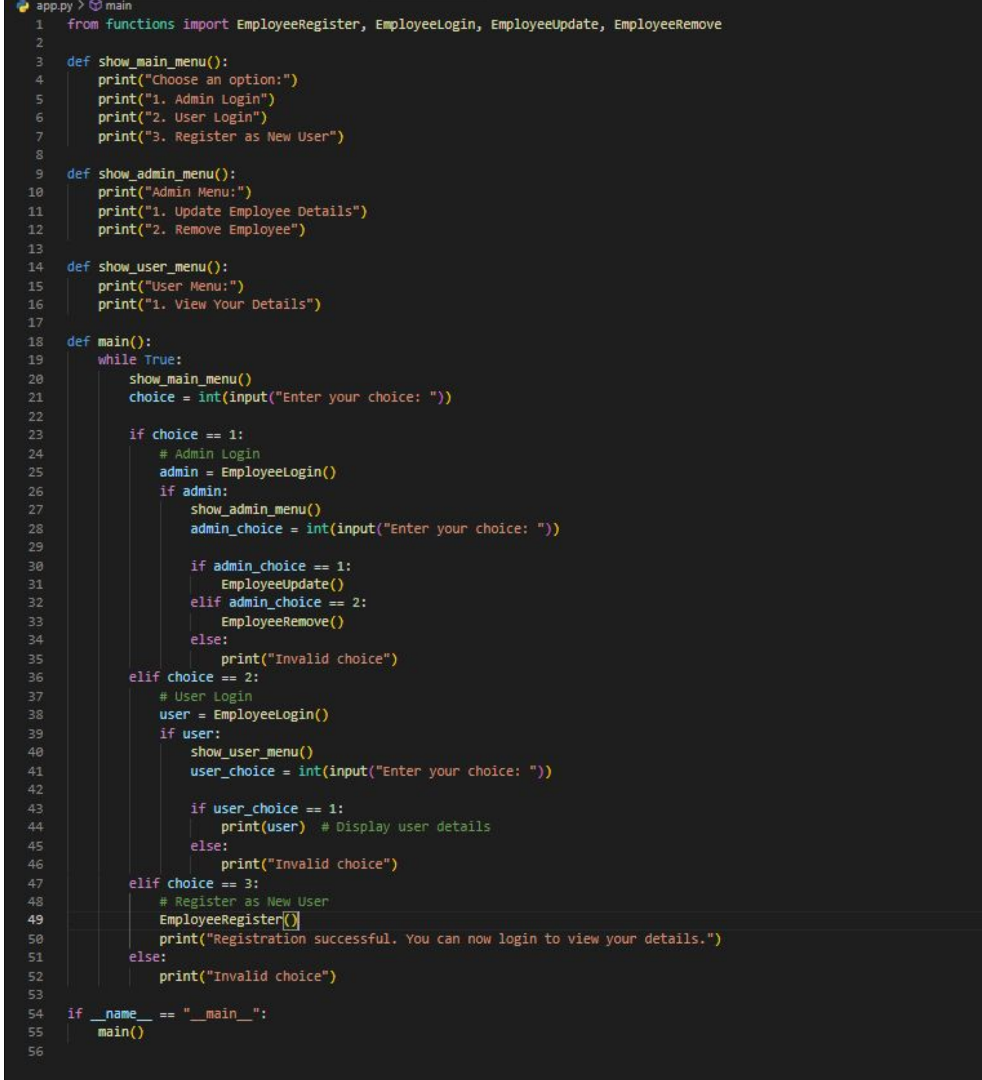
Frontend: Not applicable (CLI-based application)

Backend: Python with PostgreSQL

Database: PostgreSQL

User Interaction Flow

- **Admin Login:** Provides options for updating and removing employees.
- **Employee Login:** Provides options for viewing and updating personal details.
- **New User:** Registers and views own details.



```python
from functions import EmployeeRegister, EmployeeLogin, EmployeeUpdate, EmployeeRemove

def show_main_menu():
    print("Choose an option:")
    print("1. Admin Login")
    print("2. User Login")
    print("3. Register as New User")

def show_admin_menu():
    print("Admin Menu:")
    print("1. Update Employee Details")
    print("2. Remove Employee")

def show_user_menu():
    print("User Menu:")
    print("1. View Your Details")

def main():
    while True:
        show_main_menu()
        choice = int(input("Enter your choice: "))

        if choice == 1:
            # Admin Login
            admin = EmployeeLogin()
            if admin:
                show_admin_menu()
                admin_choice = int(input("Enter your choice: "))

                if admin_choice == 1:
                    EmployeeUpdate()
                elif admin_choice == 2:
                    EmployeeRemove()
                else:
                    print("Invalid choice")
        elif choice == 2:
            # User Login
            user = EmployeeLogin()
            if user:
                show_user_menu()
                user_choice = int(input("Enter your choice: "))

                if user_choice == 1:
                    print(user)  # Display user details
                else:
                    print("Invalid choice")
        elif choice == 3:
            # Register as New User
            EmployeeRegister()
            print("Registration successful. You can now login to view your details.")
        else:
            print("Invalid choice")

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

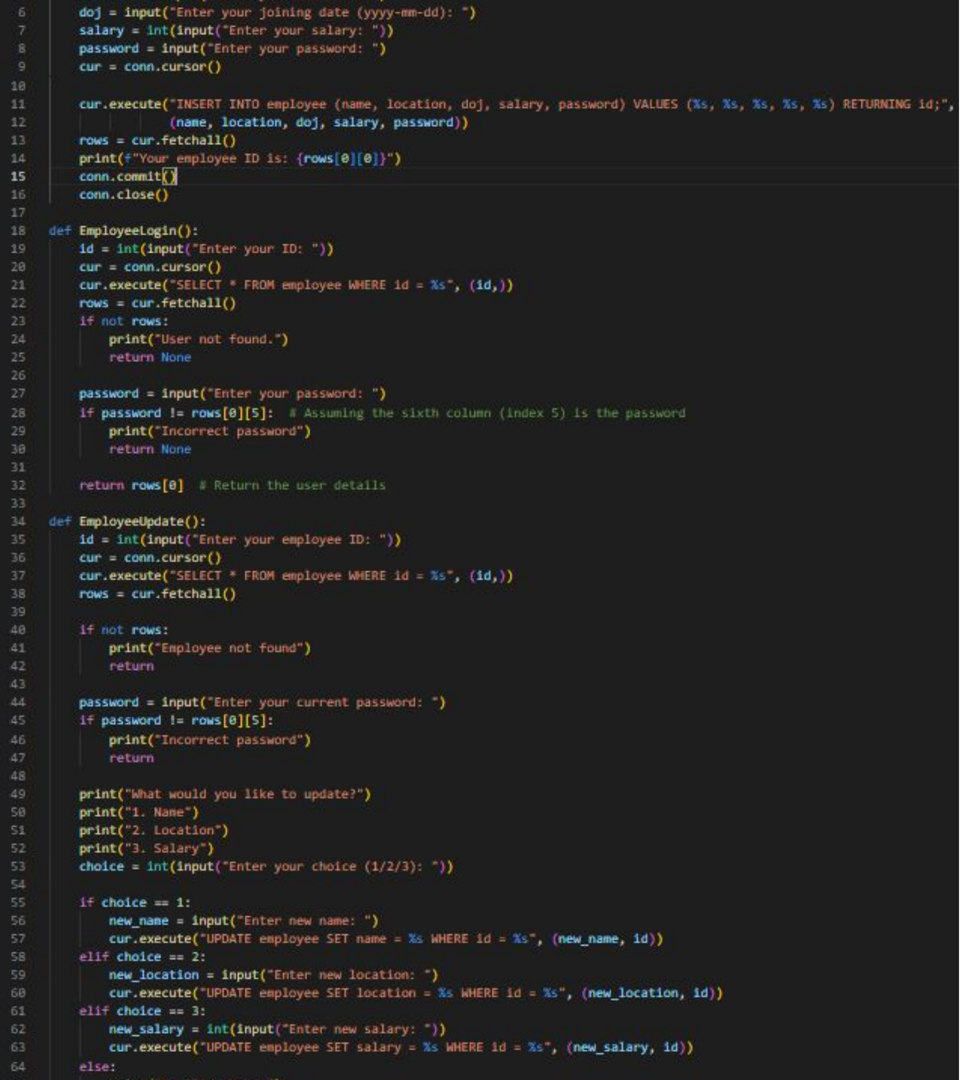
Functionalities

Registration

Login

Update

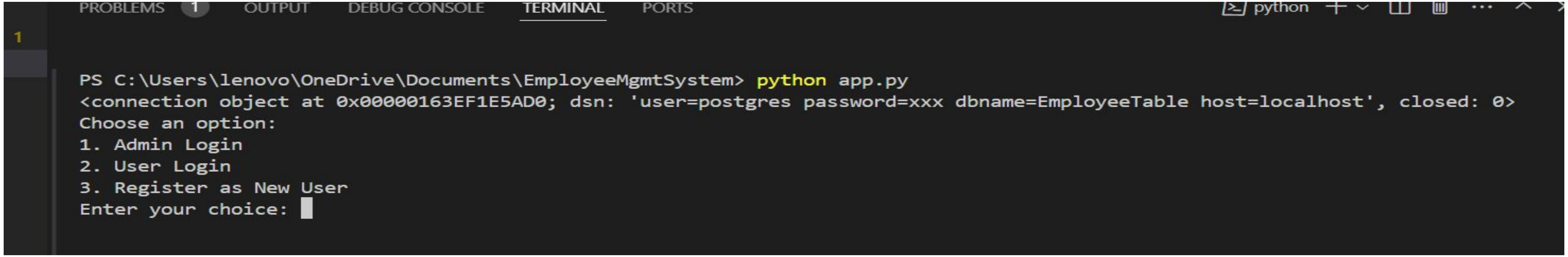
Remove



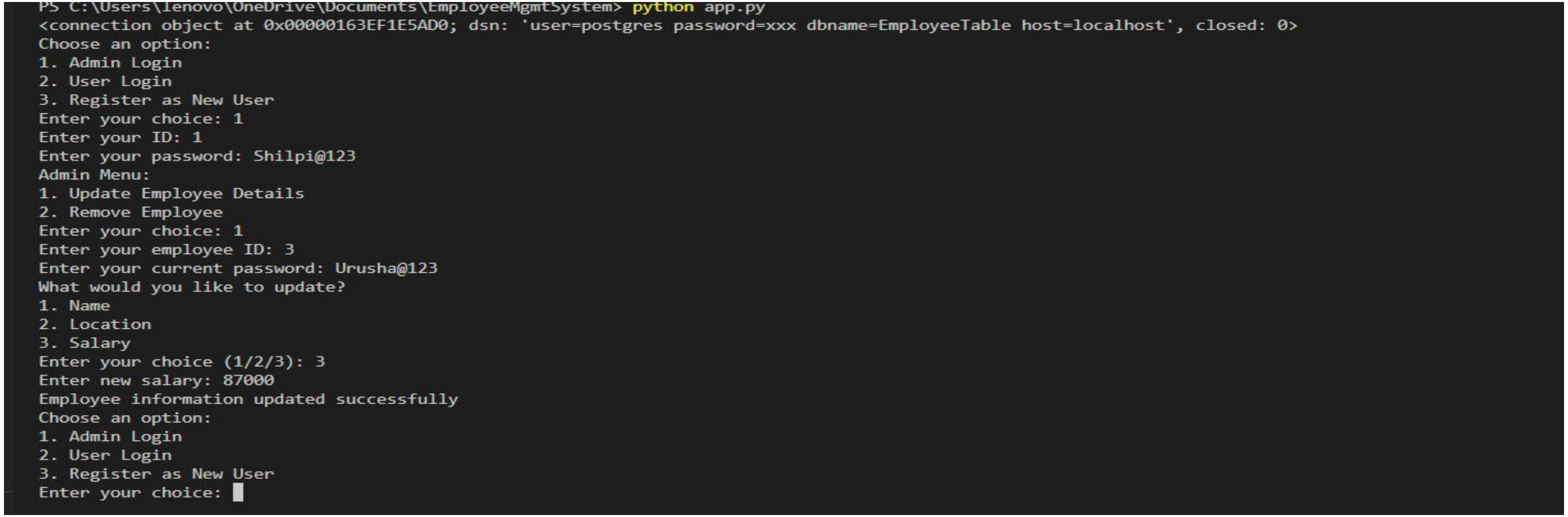
```python
    doj = input("Enter your joining date (yyyy-mm-dd): ")
    salary = int(input("Enter your salary: "))
    password = input("Enter your password: ")
    cur = conn.cursor()

    cur.execute("INSERT INTO employee (name, location, doj, salary, password) VALUES (%s, %s, %s, %s, %s) RETURNING id;",
                (name, location, doj, salary, password))
    rows = cur.fetchall()
    print(f"Your employee ID is: {rows[0][0]}")
    conn.commit()
    conn.close()

def EmployeeLogin():
    id = int(input("Enter your ID: "))
    cur = conn.cursor()
    cur.execute("SELECT * FROM employee WHERE id = %s", (id,))
    rows = cur.fetchall()
    if not rows:
        print("User not found.")
        return None

    password = input("Enter your password: ")
    if password != rows[0][5]:  # Assuming the sixth column (index 5) is the password
        print("Incorrect password")
        return None

    return rows[0]  # Return the user details

def EmployeeUpdate():
    id = int(input("Enter your employee ID: "))
    cur = conn.cursor()
    cur.execute("SELECT * FROM employee WHERE id = %s", (id,))
    rows = cur.fetchall()

    if not rows:
        print("Employee not found")
        return

    password = input("Enter your current password: ")
    if password != rows[0][5]:
        print("Incorrect password")
        return

    print("What would you like to update?")
    print("1. Name")
    print("2. Location")
    print("3. Salary")
    choice = int(input("Enter your choice (1/2/3): "))

    if choice == 1:
        new_name = input("Enter new name: ")
        cur.execute("UPDATE employee SET name = %s WHERE id = %s", (new_name, id))
    elif choice == 2:
        new_location = input("Enter new location: ")
        cur.execute("UPDATE employee SET location = %s WHERE id = %s", (new_location, id))
    elif choice == 3:
        new_salary = int(input("Enter new salary: "))
        cur.execute("UPDATE employee SET salary = %s WHERE id = %s", (new_salary, id))
    else:
```

—

Code Output



```
PS C:\Users\lenovo\OneDrive\Documents\EmployeeMgmtSystem> python app.py
<connection object at 0x00000163EF1E5AD0; dsn: 'user=postgres password=xxx dbname=EmployeeTable host=localhost', closed: 0>
Choose an option:
1. Admin Login
2. User Login
3. Register as New User
Enter your choice: 
```



```
PS C:\Users\lenovo\OneDrive\Documents\EmployeeMgmtSystem> python app.py
<connection object at 0x00000163EF1E5AD0; dsn: 'user=postgres password=xxx dbname=EmployeeTable host=localhost', closed: 0>
Choose an option:
1. Admin Login
2. User Login
3. Register as New User
Enter your choice: 1
Enter your ID: 1
Enter your password: Shilpi@123
Admin Menu:
1. Update Employee Details
2. Remove Employee
Enter your choice: 1
Enter your employee ID: 3
Enter your current password: Urusha@123
What would you like to update?
1. Name
2. Location
3. Salary
Enter your choice (1/2/3): 3
Enter new salary: 87000
Employee information updated successfully
Choose an option:
1. Admin Login
2. User Login
3. Register as New User
Enter your choice: 
```

Thank You