app.py

```
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")
    print("3. View All Details")

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()
                elif admin_choice == 3:
                    EmployeeLogin()
```

```
                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()
```

connection.py

```
import psycopg2

conn = psycopg2.connect(
    database = "EmployeeTable",
    user = "postgres",
    host = "localhost",
    password = "Shilpi@123"
)

print(conn)
```

Functions.py

```
from connection import conn

def EmployeeRegister():
    name = input("Enter your name: ")
    location = input("Enter your location: ")
    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.fetchone()
    print(f"Your employee ID is: {rows[0]}")
    conn.commit()
    cur.close()  # Make sure to close the cursor
    conn.close()


def EmployeeLogin():
    id = int(input("Enter your ID: "))
    cur = conn.cursor()

    if id == 1:
        # Admin view: Fetch all employee details
        cur.execute("SELECT * FROM employee")
        rows = cur.fetchall()  # Fetch all rows since it's an admin
viewing all employees
        if not rows:
            print("No employees found.")
            return None
        else:
            for row in rows:
```

```
                print(row)  # Print each employee's details
            return rows
    else:
        # Normal user view: Fetch only the details for the user
with the given ID
        cur.execute("SELECT * FROM employee WHERE id = %s", (id,))
        row = cur.fetchone()  # Fetch only one row since it's a
single user
        if not row:
            print("User not found.")
            return None

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

        return row  # 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.fetchone()

    if not rows:
        print("Employee not found.")
        cur.close()
        return

    password = input("Enter your current password: ")
    if password != rows[5]:  # Assuming password is in the sixth
column
        print("Incorrect password.")
        cur.close()
        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:
        print("Invalid choice.")
        cur.close()
        return

    print("Employee information updated successfully.")
    conn.commit()
    cur.close()
    conn.close()


def EmployeeRemove():
    name = input("Enter admin name: ")
    cur = conn.cursor()

    cur.execute("SELECT * FROM employee WHERE name ILIKE %s",
(name,))
    admin = cur.fetchone()

    if admin is None:
```

```
        print("Admin not found.")
        cur.close()
        return

    if admin[1] != "Shilpi":  # Assuming the second column (index
1) is the name
        print("Permission denied: Only 'Shilpi' can remove
employees.")
        cur.close()
        return

    password = input("Enter your password: ")
    if password != admin[5]:  # Assuming the sixth column (index 5)
is the password
        print("Incorrect password.")
        cur.close()
        return

    emp_id = int(input("Enter the ID of the employee you want to
remove: "))
    confirmation = input(f"Are you sure you want to remove the
employee with ID {emp_id}? (yes/no): ").lower()

    if confirmation == "yes":
        cur.execute("DELETE FROM employee WHERE id = %s",
(emp_id,))
        print(f"Employee with ID {emp_id} has been removed.")
        conn.commit()
    else:
        print("Operation canceled.")

    cur.close()
    conn.close()
```