**SRI KRISHNA COLLEGE OF**

**ENGINEERING AND TECHNOLOGY**

(An Autonomous Institution)

(Approved by AICTE and Affiliated to Anna University, Chennai)

ACCREDITED BY NBA & NAAC WITH ‘A’ GRADE

## BOOTATHON – 1

# SPEAK UP

***WHERE EVERY ONE CAN SPEAK!***

**SUBMITTED BY**

**TEAM NO-27**

RAGHUNANDAN S -18EUIT114

BALAKRISHNAN S – 18EUCS022

BALAVIGESH C – 18EUCS023

NITHISKUMAR R – 18EUCS155

MOHAMED ANSAR A – 18EUMT055

# TABLE OF CONTENTS

## ABSTRACT:

In our recent pandemic era, people are facing many difficulties to file complaints about their problems like **improper water supply, voltage fluctuations and drainage blocks,** etc. Due to the quarantine situation, public are unable to visit the municipality office and lodge their complaints. Considering the current situation, we came up with an idea of developing a **windows-based application** that solves the issue addressed above.

**SPEAK UP** – It is a windows-based application developed for the interaction between **public people and government servants**. The main motive behind this application is making the ease for public to speak against the area/locality problems and acknowledging the government about it. It also provides a dual way service by interacting to the public and resolving the issues by the government.

Our aim is to provide a **user-friendly relationship between the public(user) and the government servants(administrator)** to acknowledge the government about the public difficulties and problems in their respective localities. Once the information reaches the government, they can reply and solve the issues at the nearest future. We have covered typically three municipality problems namely water services, electricity and drainage leakages/blocks.

The idea of developing online and web services to perform such functions during the quarantine situation can have significant development on the crisis. It not only develops the communication between the common people and government workers but also **reduces the spread of pandemic disease by keeping the people safe at their homes** and hopefully bring it under control.

## EXISTING SYSTEM:

In the conventional way, people used to go to the respective offices to **complaint their problems** in their surroundings like drainage problem, cleanliness, frequent power cut and etc., and if they complaint they don’t know wheather their problem has been **solved or not** if the problem has not been solved, again the public have to complaint again in the respective offices or they forget their problem and their problem will be **unsolved and there is no solution**.

There are some online complaint organization like **MyGrievance** which is a online platform for **complaining on a particular sector for separate cities**. It is not an platform which is suitable for complaining in all cities. We won’t know wheather our complaint has been acknowledged by them or not and also we don’t know the status of our complaint.

## NOVELTY PROPOSED:

Due to this pandemic situation, Public cant go to the respective offices and file their complaint so, we came up with the idea of developing a windows based application so **that public can complaint their problems from their homes.** We also included that public can also **get informations from the government officials** like ‘This area will have a shutdown on Saturday – from Electricity sector’. It will be very useful for the public.

If a public filed a complaint the complaint list in the government official’s page can be **deleted** only if the complaint’s conformation status is **marked as solved** by both public and government officials otherwise it will always appear on the list.

Public can be able to **view the status** of their complaint wheather it is ‘Acknowledged’ or ‘Marked as solved by Government officials’.

Only the **admin can register** the Government official because to avoid that anyone can register in the name of government official.

The admin can also view the registered government official details and also the complaints filed by the public. The **admin also can’t able to delete the complaints** filed by the public if it is not marked as solved by both government official and public.

## HIGH LEVEL DOCUMENTATION:

## BUSINESS REQUIREMENTS:

The following are the business requirements for this project.

| S.NO | REQUIREMENT | STATUS |
|---|---|---|
| 1 | Login and sign-up | Implemented |
| 2 | Public Module | Implemented |
| 3 | Government Official Module | Implemented |
| 4 | Admin | Implemented |

## LIST OF FILES SUPPLIED:

| S.NO | FILE | DESCRIPTION |
|---|---|---|
| 1. | Speak_Up.java | Header File which contains all the main functions which checks for the user type logged in and redirect them to the welcome page. |
| 2. | UserType.java | Sets the user type of the Logged in user.<br>• Public User<br>• Government Official user<br>And also it is stored in the database. |
| 3. | WelcomePage.java | Welcome Page gives us an option for Login as a Public, Government Officials, Admin and also Sign Up option for Public. |
| 4. | AdminLogin.java | Where Admin can Login with his/ her Phone Number and Password. |
| 5. | AdminHomePage.java | In the Admin Home page, where the admin can register Government Officials, add admin, view the details of the registered government officials, view the complaints posted by the Public and he/she can logout from the home page. |
| 6. | AdminRegistration.java | He/she can add a new admin with his/her phone number and password and give him/her an admin access. |
| **7.** | AdminShowOfficials.java | Where the admin can view the details of the registered government officials. |
| 8. | AdminComplaints.java | Where the admin can view the list of the Complaints posted by the Public. |
| 9. | AdminShowComplaint.java | Where the admin can view the details of the complaints made by the public and the admin can delete the complaint only if it is marked as solved by both public and government officials. |
| 10. | RegistrationPublic.java | Where the Public user can register their details asked for registration like name, phone number, email-id and etc., |

|  |  |  |
|---|---|---|
| 11. | CheckValidPublic.java | Checks the validation for the entered details of public like phone number, email-id, passwords (password and confirm password) and etc., |
| 12. | RegistratinGovtServant.java | Where the admin can register the Government officials according to their sector with their details like name, phone number, password(password and confirm password), unique id and etc., |
| 13. | CheckValidGovt.java | Checks the validation for the entered details of Government officials like phone number, email-id, passwords (password and confirm password), unique id and etc., |
| 14. | LoginPage.java | Where the user can login with their given Phone number and Password. According to the user type the Login page changes and also the database.<br>• Public user<br>• Government Officials |
| 15. | PublicHomePage.java | In the Public home page, the Public user can choose options like view his/her profile, complaints, make complaints, view announcements by Government officials and also he/she can logout. |
| 16. | PublicModel.java | Where the details entered by Public user while registration process is stored. |
| 17. | PublicProfile.java | Where the details of his/her (Public user) can be viewed and they can update their profile, change their password and also he can delete his/her account permanently. |
| 18. | PublicComplaints.java | In the public complaints page, he/she (public user) can make complaint according to the sector and by giving the details of the complaint. |

|  |  |  |
|---|---|---|
| 19. | ShowMyComplaint.java | In the show my complaints page, the public user can view the details of the complaint that he/she made and also view the status of their complaint.<br>• Acknowledged<br>• Solved<br>And also he/she can delete his complaint once it is solved by the government official. |
| 20. | PublicInformation.java | In the public information page, the user can view the list of announcements made by the Government officials according to their sectors. |
| 21. | ShowAnnouncement.java | In the show announcements, the public user can view the brief details of the announcements made by the government officials according to their sectors. |
| 22. | GovtHomePage.java | In the Government Official home page, the Government official can view his/her profile, view his/her posted information, post information to the public, view complaints posted by the public and also he/she can logout. |
| 23. | GovtModel.java | Where the details entered by Admin while registering the Government officials is stored. |
| 24. | GovtProfile.java | Where the details of his/her (Government Officials) can be viewed and they can update their profile, change their password and also he can delete his/her account permanently. |
| 25. | GovtInformation.java | In the government information page, the Government official can post the information of their sector which can be viewed by the public. |
| 26. | MyInformations.java | In my information page, the government official can view the list of information that he/she posted to the public. |

| | | |
|---|---|---|
| 27. | ShowPostedInfo.java | In the show posted information page, the Government officials can view the details of the information that he/she posted to the public. And also he/she can delete the information posted to the public. |
| 28. | GovtComplaints.java | In the government Complaints page, the government official can view the list of complaints posted by the public in their sector. |
| 29. | ShowComplaint.java | In the show complaints page, the government official can view the details of the complaint posted by the public and he/she can give the status that their complaint has been<br>• Acknowledged<br>• Solved<br>Once it is marked as solved by both the Government official and the public, the complaint can be deleted by the government officials otherwise it can't be deleted. |
| 30. | ChangePassword.java | Where the user can change their Current password with their new password. |
| 31. | HashPassword.java | Hash password is used to store the password in hexadecimal for the security reasons. |
| 32. | DbConnection.java | In the DB Connection file, the connectivity to database (PostgreSQL) is written. |

### WELCOME PAGE:

The project begins with a welcome page as shown below. In the **WELCOME PAGE**, the application has separate login buttons for users of **public and government officials** to separate the type of users logging in our application and a separate signup option for public. It also contains a separate login option for administrator.

### LOGIN PAGE – ADMIN:

This page requires the administrator **phone number and password** to login into the admin’s account. The ‘Show Password’ checkbox is also available to see whether the typed password is correct or not. If the given phone number and password matches with the administrator database, it will redirect to the Admin Home page.

## HOME PAGE – ADMIN:

The page begins with a welcome note and a **logout option** at the right top corner. The admin **can register a government servant** or **add another admin** using the available buttons and also he/she can view all the registered government official details and the complaints made by the public.

## REGISTRATION PAGE – ADMIN:

The administrator registration requires the government verification whether he is an **employee of government**. The phone number and password created by the higher authority person of government is required to access the **HOME page of government admin**. The ‘Submit’ button must be clicked once the details are entered. The ‘Go Back’ button is also available to return to the previous page.

## SHOW OFFICIALS PAGE – ADMIN:

Where the admin can view the **details of the registered government officials** and the admin can separate the government by sector and city if the admin wants.

## SHOW COMPLAINTS PAGE – ADMIN:

Where the admin can view the **list of the Complaints** posted by the Public.

## SHOW DETAILED COMPLAINT – ADMIN:

Where the admin can view the details of the complaints made by the **public** and the admin can **delete** the complaint only if it is **marked as solved** by both public and government officials.

## REGISTRATION PAGE – GOVERNMENT OFFICIAL:

The administrator can register their account by entering the required fields namely name, phone number, email-id, sector, city, Aadhar number, password. If the required fields are valid, it will be stored in the database or the user(government) will be asked to alter the invalid fields. The password is stored as **hash password in hexadecimal format to ensure security.**

We have included the mentioned validations for this module:

- Password should contain a minimum of 8 characters with an uppercase, lowercase and a number.
- Phone number should be of 10 digits and it should start from [6-9].
- Aadhar number should be of 12 digits.
- Email ID should contain an alphabet, period(.) and address character (@) and can also contain with numbers.
- All fields are required to be entered for registration.
- Both the password fields (password and confirm password) should match to create it.

## REGISTRATION PAGE – PUBLIC:

The public users can register their account by entering the required fields namely name, e-mail, gender, password, address, area, city, pin code and phone number. If the required fields are valid, it will be stored in the database or the user(public) will be asked to alter the invalid fields. The password is stored as **hash password in hexadecimal format to ensure security.**

We have included the mentioned validations for this module:

- Password should contain a minimum of 8 characters with an uppercase, lowercase and a number.
- Phone number should be of 10 digits and it should start from [6-9].
- Pin code should be of 6 digits.
- Email ID should contain an alphabet, period(.) and address character (@). It can also contain numbers.
- All fields are required to be entered for registration.
- Both the password fields (password and confirm password) should match to create it.

## LOGIN PAGE – PUBLIC:

The public user who signed up in our application can login with his/her **phone number and password**, which he/she used to create the account. In this page, the validation of registered phoned number and password is **checked with the database** details. There is also a separate button for unregistered users to create an account using **'sign up as public'.**

## LOGIN PAGE – GOVERNMENT OFFICIAL:

The government official who signed up in our application by **Administrator** can login with his/her **phone number and password**, which he/she used to create the account. In this page, the validation of registered phoned number and password is **checked with the database** details**.**

## PUBLIC HOME PAGE:

When a public user has logged in successfully, he/she will be redirected to the public home page. In this page, public user will be able to view the profile using PROFILE button and can also edit the profile using EDIT button. The changes will be saved to the database. LOGOUT button is used to log out the public user and redirect to the welcome page. The public user can also **post complaints and receive information** about the progress of solution in this page. The user can also view his/her **previous complaints posted earlier.**

## PUBLIC PROFILE PAGE:

In the public PROFILE module, we can see the information of the public user who has logged in using the public login page. We can **update** the information of the public user by clicking EDIT button and we can also **change the password and delete the user account.**

## PUBLIC MAKE COMPLAINT PAGE:

The public user can **file a complaint** about the municipality problems in his/her locality. The user must mention the sector, area (default), a short description with the brief information about the complaint.

## PUBLIC SHOW ANNOUNCEMENT PAGE:

In this page, the Information posted by the Government Officials will be displayed, **Information relevant to the Public's area** only displayed. We can view the detailed information by clicking view button.

BACK button at the bottom will redirect the Public to their HOME PAGE.

## PUBLIC SHOW DETAILED ANNOUNCEMENT PAGE:

In this page, the public can view the **detailed announcement of the posted information** of the government official according to their sector.

## PUBLIC SHOW MY COMPLAINT PAGE:

The government employee can check the various complaints posted by the public users in **‘My Complaints’** tab and view the details for each using the ‘View’ button.

## PUBLIC SHOW DETAILED MY COMPLAINT PAGE:

The government user can view the detailed complaint of a public user as shown below. It includes the status, city, area, posted date, sector, description and information. The government employee can also click the **‘Solved’** checkbox is the problem is resolved and **delete the complaint** if it is not legitimate.

## GOVERNMENT OFFICIALS POST ANNOUNCEMENT PAGE:

The POST INFORMATION page has three section – **SECTOR, AREA, INFORMATION.** SECTOR shows the SECTOR in which the Government Official works which is default and it can't be edited because it will be taken from the PROFILE of Government Officials so that they can't change their sector.

AREA shows the areas under the City select to which area the information to be posted. INFORMATION is the message that should be posted as an information to the Public.

POST button at the bottom will be used to post the information to the Public (i.e.) **Information will be stored in the database.**

BACK button at the bottom will redirect the Government Officials to their HOME PAGE.

## GOVERNMENT OFFICIALS MY ANNOUNCEMENT PAGE:

In the Posted Information Page, the Government Officials can able to view the list of information posted by them to the Public.

This page contains three columns **- POSTED AT, AREA, VIEW**

POSTED AT shows the date at which the information is posted

AREA shows to which area the information is posted, and VIEW section contains the VIEW button, by clicking that the Government Officials will be redirected to the DETAILED INFORMATION PAGE.

BACK button at the bottom will redirect the Government Officials to their HOME PAGE.

## GOVERNMENT OFFICIAL VIEW DETAILED INFORMATION PAGE:

In this information page, the Government Officials can be able to view the detailed information posted to the Public like to which **Area, City and the Sector** in which the information is posted to the Public.

The Government Officials can able to delete the information posted to the Public, by using the **DELETE** button at the bottom and, BACK button will redirect to the GOVERNMENT VIEW INFORMATION PAGE.

## GOVERNMENT OFFICIALS VIEW COMPLAINTS PAGE:

In this page, the list of complaints posted by the Public will be displayed here. It contains – **NAME, SECTOR, CITY, AREA, VIEW.**

NAME displays the Public’s name who posted the complaint.

SECTOR displays the sector in which there is a complaint

CITY and AREA displays the Public’s area and city in the VIEW section, the VIEW button will redirect the Government Officials to the Detailed Complaint Page

BACK button at the bottom will redirect the Government Officials to their HOME PAGE.

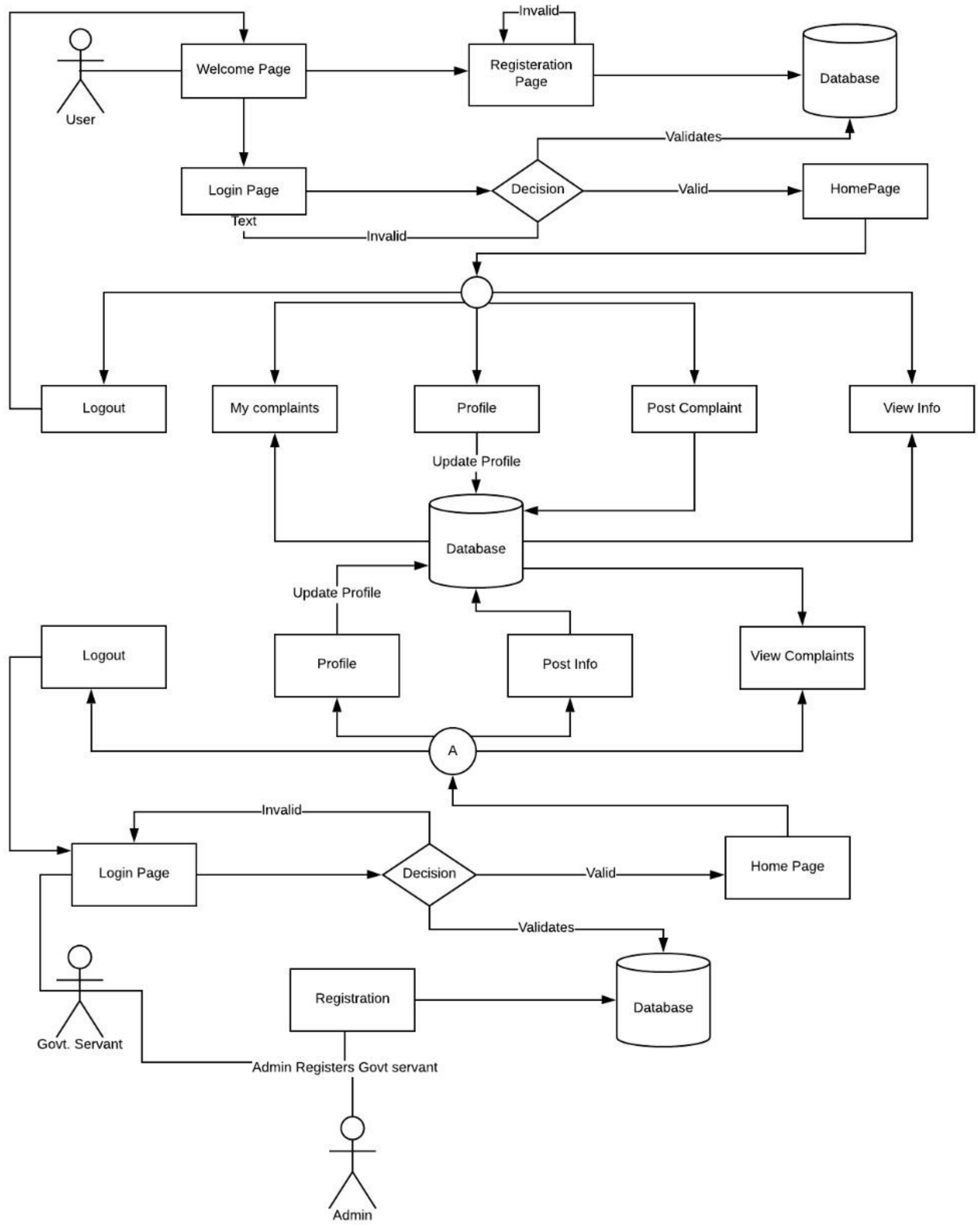
## GOVERNMENT OFFICIALS VIEW DETAILED COMPLAINTS PAGE:

In this complaint page, the Government Officials can be able to view the detailed complaint posted by the Public. It contains Public’s Name, City, Area, Phone Number, and the detailed information about the complaint posted by the Public.

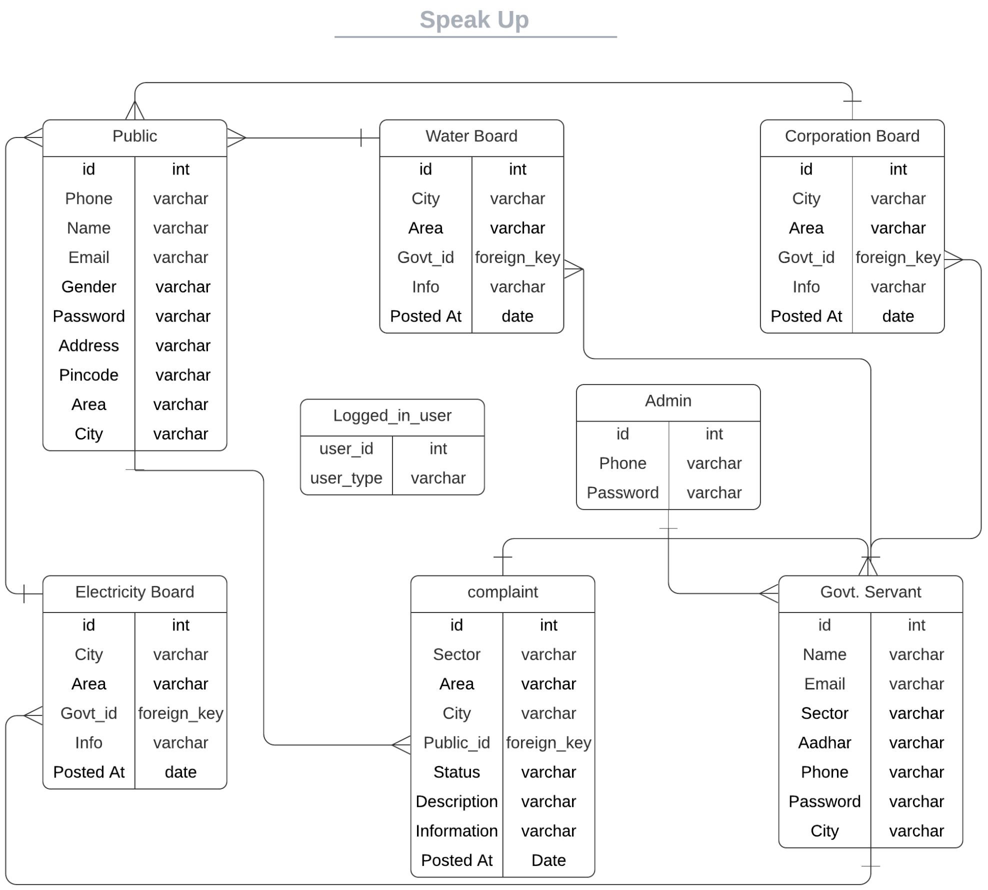
The **STATUS** of the Public’s Problem will be updated here.

# LOW LEVEL DOCUMENTATION:

# DATA FLOW DIAGRAM:



# DATABASE DESIGN:



| Public | |
|---|---|
| id | int |
| Phone | varchar |
| Name | varchar |
| Email | varchar |
| Gender | varchar |
| Password | varchar |
| Address | varchar |
| Pincode | varchar |
| Area | varchar |
| City | varchar |

| Water Board | |
|---|---|
| id | int |
| City | varchar |
| Area | varchar |
| Govt_id | foreign_key |
| Info | varchar |
| Posted At | date |

| Corporation Board | |
|---|---|
| id | int |
| City | varchar |
| Area | varchar |
| Govt_id | foreign_key |
| Info | varchar |
| Posted At | date |

| Logged_in_user | |
|---|---|
| user_id | int |
| user_type | varchar |

| Admin | |
|---|---|
| id | int |
| Phone | varchar |
| Password | varchar |

| Electricity Board | |
|---|---|
| id | int |
| City | varchar |
| Area | varchar |
| Govt_id | foreign_key |
| Info | varchar |
| Posted At | date |

| complaint | |
|---|---|
| id | int |
| Sector | varchar |
| Area | varchar |
| City | varchar |
| Public_id | foreign_key |
| Status | varchar |
| Description | varchar |
| Information | varchar |
| Posted At | Date |

| Govt. Servant | |
|---|---|
| id | int |
| Name | varchar |
| Email | varchar |
| Sector | varchar |
| Aadhar | varchar |
| Phone | varchar |
| Password | varchar |
| City | varchar |

## ALGORITHM:

### WELCOME PAGE:

```
1.Start
2.Open welcome page.
3.Select any of the four available options.
            - Login as Public.
            - Login as Government official.
            - Login as Admin.
            - Sign Up as Public.
4.Stop
```

### LOGIN PAGE – ADMIN:

```
1.Start
2.Open Admin login module,
3. Get input (Phone Number, Password)
4. If selected button is “Submit"
      check if Phone Number and Password is in the database(Admin)
            then go to "Admin Home page module"
      else
            then go to "Admin login module"
  else if selected button is "Back"
      then go to "Welcome page module"
5.Stop
```

## LOGIN PAGE – PUBLIC:

```
1.Start
2.Open login page module,
3. Get input (Phone Number, Password)
4. If selected button is “Submit"
         check if Phone Number and Password is in the database(public)
                  then go to "Public Home page module"
         else
                  then go to "Login page module"
   else if selected button is "Back"
         then go to "Welcome page module"
   else if selected button is "Sign up as public"
         then go to "Registration public module"
5.Stop
```

## LOGIN PAGE – GOVERNMENT OFFICIAL:

```
1.Start
2.Open login page module,
3. Get input (Phone Number, Password)
4. If selected button is “Submit"
         check if Phone Number and Password is in the database(Government)
                  then go to "Government Home page module"
         else
                  then go to "Login page module"
   else if selected button is "Back"
         then go to "Welcome page module"
5.Stop
```

**REGISTRATION PAGE – ADMIN:**

```
1.Start

2.Open Registration Admin module

3.Get input (Phone Number, Password)

4.if selected button is “Register”
        if (values validation==true)
            then update the database (Admin) with new values
            then go to “Admin Login module”
        else
            then go to “Public Registration module”
    else if selected button is “Back”
        then go to “Admin Home Page module”

5.Stop
```

**REGISTRATION PAGE – PUBLIC:**

```
1.Start

2.Open Registration Public module

3.Get input (Name, Email-ID, Gender, Password, Confirm Password, Address, Area, City,
Pin code, Phone Number)

4.if selected button is “Register”
        if (values validation==true)
            then update the database (Public) with new values
            then go to “Public Login module”
        else
            then go to “Public Registration module”
    else if selected button is “Back”
        then go to “Welcome Page module”

5.Stop
```

## REGISTRATION PAGE – GOVERNMENT OFFICIALS:

```
1. Start
2. Open Registration Government Servant module,
3. Get input (Name, Phone Number, Email-ID, Sector, City, Aadhaar Number, Password,
Confirm Password)
4. If selected button is “Register"
        check if the values for validation
                then update the database(Government) with new values
        else
                then go to "Registration Government Servant module"
   else if selected button is "Back"
        then go to "Admin login module"
5. Stop
```

## HOME PAGE – ADMIN:

1.Start

2.Open admin home page.

3.Select any of the three available options.

- Register Government officials.
- View Government Officials
- View Public Complaints
- Add admin.
- Logout.

4.Stop

## HOME PAGE – PUBLIC:

1.Start

2.Open public home page.

3.Select any of the five available options.

- View Announcements.
- Make Complaints.
- Go to Profile.
- View Posted complaints.
- Logout.

4.Stop

## HOME PAGE – GOVERNMENT OFFICIALS:

```
1.Start
2.Open government official home page.
3.Select any of the five available options.
            - Post Announcements.
            - View Complaints.
            - Go to Profile.
            - View Posted announcements.
            - Logout.
4.Stop
```

## PROFILE PAGE – PUBLIC:

```
1.Start
2.Open Public Profile module
3.If selected button is “Edit”
               then edit the profile of the user (Public)
               if selected button is “Save”
                         then update the database with new values
               else if selected button is “Cancel”
                         then go to “public profile module”
       else if selected button is “Change Password”
               then go to “Change Password module”
       else if selected button is “Delete Account”
               then remove the user from database
               then go to “Welcome Page module”
       else if selected button is “Back”
               then go to “Public Home module”
4.Stop
```

**PROFILE PAGE – GOVERNMENT OFFICIALS:**

```
1.Start
2.Open the government profile module,
3. If selected button is “Edit"
            then edit the profile of the user (government servant)
            If selected button is “Save"
                        then update the database with new values
            else if selected button is "Cancel"
                        then go to "government profile module"
    else if selected button is "Change Password"
            then go to "Change Password module"
   else if selected button is "Delete Account"
            then remove the user from database
            then go to "Welcome page module"
    else if selected button is "Back"
            then go to "Government Home module"
4.Stop
```

## CHANGE PASSWORD – (PUBLIC/GOVERNMENT OFFICIALS):

```
1. Start
2. Open Change password module,
3. Get input (Current password, New password, Confirm password)
4. If selected button is “Change"
            check if the values for validation
                        then update the database(Public/Government) with new values
            else
                        then go to "Change password module"
   else if selected button is "Cancel"
            then go to "(Public / Government) Profile page module"
5. Stop
```

## VIEW ANNOUNCEMENTS PAGE – PUBLIC:

```
1.Start
2.Open Show Announcements module
3.If selected button is “View”
                then go to Public Information module
   else if selected button is “Back”
                then go to Public Home Page module
4.Stop
```

## MAKE COMPLAINT PAGE – PUBLIC:

```
1.Start
2.Open Make Complaint Page.
3.Select Sector
4.Input the Description, Information for the selected area.
5.if (Submit is clicked)
        if (information not null)
                Add Message to Database
                Show posted message
        else
                Show error message
6.if (back is clicked)
                Go to Home Page
7.Stop
```

## MY COMPLAINTS PAGE – PUBLIC:

```
1.Start.
2.Open My Complaints Page.
3.List all the complaints with the sector, city and posted at date of the complaint.
4.Click on view of any of the listed complaints to view all the details of the complaints in the Show My Complaint Page
        1.Start
        2.Open Show My Complaints Page.
        3.Show detailed information about the complaint.
        4.Show the status of the complaint.
        5.Acknowledge the complaint or Mark the complaint as solved by the Government Officials.
        6.If Marked as acknowledged by the government official
            then status changed as Acknowledged.
        7.If Marked as solved by the government official
            then status changed as Pending
        8.If Marked as solved by the Public user
            then status changed as Solved
        9.if delete is clicked
            Remove Complaint from Database
        10.if Back is clicked
            Show listed My complaints page
        11.Stop.
5.Repeat step 4 for all the complaints.
6.if (Back is clicked)
        Show Home page
7.Stop.
```

## POST ANNOUNCEMENT – GOVERNMENT OFFICIALS:

```
1.Start
2.Open Post Announcement Page.
3.Select Area
4.Input the Information for the selected area.
5.if (Post is clicked)
        if (information not null)
                Add Message to Database
                Show posted message
        else
                Show error message
6.if (back is clicked)
                Go to Home Page
7.Stop
```

## MY ANNOUNCEMENTS – GOVERNMENT OFFICIALS:

```
1.Start
2.Open Show Announcement module,
3. Show the information posted by the user(Government)
4.If selected button is "Back"
          then go to "Posted information page module"
   else if selected button is "Delete"
          then remove information from database(Government)
5.Stop
```

### VIEW COMPLAINTS – GOVERNMENT OFFICIALS:

```
1.Start.
2.Open View Complaints Page.
3.List all the complaints in the corresponding sector and city of the official.
4.Click on view of any of the listed complaints to view all the details of the complaints in the Show Complaint Page

        1.Start
        2.Open Show Complaints Page.
        3.Show detailed info about the complaint.
        4.Show the status of the complaint.
        5.Acknowledge the complaint or Mark the complaint as solved.
        6.If Marked as solved by the person posted the complaint
            then enable delete for the complaint.
        7.if delete is clicked
            Remove Complaint from Database
        8.if Back is clicked
            Show listed complaints page
        9.Stop.

5.Repeat step 4 for all the complaints.
6.if (Back is clicked)
        Show Home page
7.Stop.
```

## VIEW COMPLAINTS – ADMIN:

```
1.Start.

2. Open View Complaints Page.

3. List all the complaints in all sectors and cities.

4. Click on view of any of the listed complaints to view all the details of the complaints in the Show Complaint Page

        1.Start
        2.Open Show Complaints Page.
        3.Show who made the complaint
        4.Show which official is in-charge.
        5.Show the status of the complaint.
        6.if delete is clicked

            if (status is solved)

                Remove Complaint from Database

        7.if Back is clicked

            Show listed complaints page

        8.Stop.

5.Repeat step iv for all the complaints.

6.if (Back is clicked)

        Show Home page

7.Stop.
```

## VIEW OFFICIALS – ADMIN:

```
1.Start
2.Open View Officials.
3.Display all the officials from all cities and all sectors.
4.if (sector combo box changed)
        if (city combo box selected index is greater than one)
            Show the officials in the corresponding sector and city
        else
            Show all the officials in the selected sector.
5.if (city combo box changed)
        if (sector combo box selected index is greater than one)
            Show the officials in the corresponding sector and city
        else
            Show all the officials in the selected city.
6.if (Back is clicked)
        Show home page
7.Stop
```

## PSEUDO CODE:

### WELCOME PAGE:

```
Begin
        Open Welcome page.
        Click on a button
        if (Login as Public == Clicked)
                Show Public Login Page
        else if (Login as government == Clicked)
                Show Government Login Page.
        else if (Sign up as public == Clicked)
                Show Registration Public Page.
End
```

### LOGIN PAGE – ADMIN:

```
Begin
        Open Admin login page
        input (Phone Number, Password)
        Click on a button
        if (Submit == Clicked)
                if (input Phone Number, Password==Phone Number, Password in database)
                        Show Admin home page
                else
                        Show Admin login page
        else if (Back == Clicked)
                Show Welcome page
End
```

**LOGIN PAGE – PUBLIC:**

```
Begin
    Open login page
    input (Phone Number, Password)
    Click on a button
    if (Submit == Clicked)
        if (input Phone Number, Password==Phone Number, Password in database)
            Show Public home page
        else
            Show login page
    else if (Back == Clicked)
        Show Welcome page
    else if (Sign up as Public== Clicked)
        Show Registration public page
End
```

**LOGIN PAGE – GOVERNMENT OFFICIAL:**

```
Begin
    Open login page
    input (Phone Number, Password)
    Click on a button
    if (Submit == Clicked)
        if (input Phone Number, Password==Phone Number, Password in database)
            Show Government home page
        else
            Show login page
    else if (Back == Clicked)
        Show Welcome page
End
```

**REGISTRATION PAGE – ADMIN:**

```
Begin
      Open Admin login page
      input (Phone Number, Password)
      Click on a button
      if (Submit == Clicked)
            if (input Phone Number, Password==Phone Number, Password in database)
                  Show Admin home page
            else
                  Show Admin login page
      else if (Back == Clicked)
            Show Welcome page
End
```

**REGISTRATION PAGE – PUBLIC:**

```
Begin
     Open Registration Public module
     Add the required input values for registration
     if (Register is clicked)
          if(values validation==true)
               Update the new user(Public) in the database(Public)
                Redirect to Public Login module
          else
                Show Registration Public module with error message
     else if (Back is clicked)
          Redirect to Welcome Page module
End
```

**REGISTRATION PAGE – GOVERNMENT OFFICIALS:**

```
Begin
    Open Government registration page.
    input (Name, Phone Number, Email-ID, Sector, City, Aadhar Number, Password, Confirm Password)
    Click on a button
    if (Register == Clicked)
        updates the values in database(government)
        show Welcome page
    else if(Back==Clicked)
        show Admin home page
End
```

**HOME PAGE – ADMIN:**

```
Begin
    Open Admin home page.
    Click on a button
    if (Register Government Official == Clicked)
        Show Register Government Official Page.
    else if (Add Admin == Clicked)
        Show Add Admin Registration page.
    else if (View complaints == Clicked)
        Show View Government officials page.
    else if (View Government officials == Clicked)
        Show View Government officials page.
    else if (Logout == Clicked)
        Logout and show Welcome Page.

End
```

**HOME PAGE – PUBLIC:**

```
Begin
        Open government official home page.
        Click on a button
        if (View Announcement == Clicked)
                Show View Announcement Page
        else if (Make Complaints == Clicked)
                Make Complaints Page.
        else if (Profile == Clicked)
                Show Profile Page.
        else if (View My Complaints == Clicked)
                Show My Complaints page.
        else if (Logout == Clicked)
                Logout and show Welcome Page.
End
```

**HOME PAGE – GOVERNMENT OFFICIALS:**

```
Begin
        Open government official home page.
        Click on a button
        if (Post Announcement == Clicked)
                Show Post Announcement Page
        else if (View Complaints == Clicked)
                Show Complaints Page.
        else if (Profile == Clicked)
                Show Profile Page.
        else if (View posted Announcements == Clicked)
                Show Posted announcements page.
        else if (Logout == Clicked)
                Logout and show Welcome Page.
End
```

**PROFILE PAGE – PUBLIC:**

```
Begin
    Open Public Profile module
    if (Edit is clicked)
            update the new values in text field
            if (values validation==true)
                update new values in database (Public)
            else
                show Public Profile module with error message
    else if (Change Password is clicked)
            Show Change Password module
    else if (Delete Account is clicked)
            delete profile from database (Public)
            Show Welcome Page
    else if (Back is clicked)
            Show Public Home Page
End
```

**PROFILE PAGE – GOVERNMENT OFFICIALS:**

```
Begin
    Open government profile page.
    Click on a button
    if (Edit == Clicked)
        update the new values in the text field
        if (Edit == Clicked)
            check if the values for validation
            then update the database(Government) with new values
        else
            Show Government profile page
    else if (Change Password == Clicked)
        Show Change Password Page.
    else if (Delete Account == Clicked)
        remove profile from database
        Show Welcome page.
    else if (Back == Clicked)
        Show Government home page.
End
```

**CHANGE PASSWORD – (PUBLIC/GOVERNMENT OFFICIALS):**

```
Begin
        Open change password page
        input (Current password, New password, Confirm password)
        Click on a button
        if (Change == Clicked)
                if(input Current Password==Current Password in database)
                        update password in database
                        Show (public/government) profile page
                else
                        Show change password page
        else if (Back == Clicked)
                Show (public/government) profile page

End
```

**VIEW ANNOUNCEMENTS PAGE – PUBLIC:**

```
Begin
      Open View Announcements module
      List all announcements corresponding to the Public
       if (View is clicked)
               Redirect to Public Information module
                Show detailed announcement
       else if (Back is clicked)
               Redirect to Public Home Page module
End
```

**MAKE COMPLAINT PAGE – PUBLIC:**

```
Begin
        Open Make Complaints Page
        Input Sector
        Input Description, Information for the selected area
        if (Submit == Clicked)
                if (Information! = Null)
                        Add Information to Database
                        Show Submit message
                else
                        Show error message
        if (Back == Clicked)
                Show Public Home Page
End
```

**MY COMPLAINTS PAGE – PUBLIC:**

```
Begin
    Open My Complaints Page
    List all complaints corresponding to the Sectors you complained
    if (View is clicked)
        Show detailed information.
        Click on any corresponding action.
        if (Acknowledge == Clicked in Government Official's complaint page)
             Complaint's status = Acknowledged.
        else if (Solved == Clicked in Government Official's complaint page)
            Complaint's status = Pending
        else if (Marked as Solved by public)
                Complaint's status = Solved
        else if (Delete == clicked)
            Remove the complaint from database
        else if (Back == Clicked)
            Show Listed Complaints Page
    Repeat for all complaints
    if (Back == Clicked)
        Show Home Page
End
```

**POST ANNOUNCEMENT – GOVERNMENT OFFICIALS:**

```
Begin
    Open Post Announcement Page
    Input Area
    Input Information for the selected area
    if (Post == Clicked)
        if (Information! = Null)
            Add Information to Database
            Show Posted message
        else
         else
            Show error message
    if (Back == Clicked)
        Show Government Official Home Page
End
```

**MY ANNOUNCEMENTS – GOVERNMENT OFFICIALS:**

```
Begin
    Open Show Announcement page
    show the information posted
    Click on a button
    if (Delete == Clicked)
        remove information from database(Government)
        show Posted information page
    else if(Back==Clicked)
        show Posted information page

End
```

**VIEW COMPLAINTS – GOVERNMENT OFFICIALS:**

```
Begin
    Open View Complaints Page
    List all complaints corresponding to the official
    if (View is clicked)
        Show detailed info.
        Click on any corresponding action.
        if (Acknowledge == Clicked)
            Complaint’s status = Acknowledged.
        else if (Solved == Clicked)
            Complaint’s status = Pending
            if (Marked as Solved by public)
                Complaint’s status = Solved
                delete button = Enabled
        else if (Delete == clicked)
            Remove the complaint from database
        else if (Back == Clicked)
            Show Listed Complaints Page
    Repeat for all complaints
    if (Back == Clicked)
        Show Home Page
End
```

## VIEW COMPLAINTS – ADMIN:

```
Begin
    Open View Complaints Page
    List all complaints.
    if (View is clicked)
        Show detailed info about who made the complaint and who is responsible.
        Show the status of the complaint.
        if (status == solved)
            delete enabled
        else if (Delete == clicked)
            Remove the complaint from db
        else if (Back == Clicked)
            Show Listed Complaints Page
    Repeat for all complaints
    if (Back == Clicked)
        Show Home Page
End
```

## VIEW OFFICIALS – ADMIN:

```
Begin
    Open View Officials
    List all officials.
    if (Sector Combo Box == Changed)
        if (City Combo Box index > 0)
            List official in selected city and sector
        else
            List all officials in selected Sector
    if (City Combo box == Changed)
        if (Sector Combo Box index > 0)
            List official in selected city and sector
        else
            List all officials in selected city.
    if (Back == clicked)
        Show Home page
End
```

## SOURCE CODE:

## HEADER FILE:

```
package speak_up;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Speak_Up {

    public static void main(String[] args) {
        Connection con;
        try {
            con = DbConnection.getConnection();
            Statement stmt = con.createStatement();
            String query = "select * from logged_in_user";
            ResultSet logged_user = stmt.executeQuery(query);
            if(logged_user.next()){
                int user_id = logged_user.getInt("user_id");
                String user_type = logged_user.getString("user_type");
                if(user_type.equals("public")) {
                    String query2 = "select * from public where id='"+user_id+"'";
                    ResultSet user = stmt.executeQuery(query2);
                    user.next();
                    PublicModel.login(user);
                    UserType.setUserType("Public");
                    PublicHomePage p1 = new PublicHomePage();
                }
                else {
                    String query2 = "select * from govt_servant where id='"+user_id+"'";
                    ResultSet user = stmt.executeQuery(query2);
                    user.next();
                    GovtModel.login(user);
                    UserType.setUserType("Govt. Servant");
                    GovtHomePage g1 = new GovtHomePage();
                }
            }
            else {
                WelcomePage w1 = new WelcomePage();
            }
            con.close();
        }
        catch(SQLException ex) {
            System.out.println("Some error occurred while setting the user model " + ex);
        }
    }
}
```

## SET USER TYPE:

```
package speak_up;

public class UserType {

    static String userType;

    static void setUserType(String type) {
        userType = type;
    }
}
```

## WELCOME PAGE:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.border.LineBorder;

public class WelcomePage extends JFrame {

    public WelcomePage()
    {
        setTitle("Speak Up");
        setSize(570, 500);
        setBounds(350, 100, 570, 500);

        Font headerfont = new Font("Itc New Baskerville",Font.BOLD,13);

        Color c1=new Color(50, 105, 168);
        Color c2=new Color(0,0,0);
        Color c3=new Color(255, 255, 255);

        JPanel panel_one = new JPanel();
        panel_one.setBackground(c3);
        panel_one.setBounds(0, 0, 570, 500);
        panel_one.setLayout(null);

        JPanel MainPanel = new JPanel();
        MainPanel.setBackground(c3);
        MainPanel.setBounds(400, 10, 150, 150);
        MainPanel.setLayout(null);

        JLabel lbl_more1 = new JLabel("___");
        JLabel lbl_more2 = new JLabel("___");
        JLabel lbl_more3 = new JLabel("___");
        lbl_more1.setBounds(80, 1, 30, 15);
        lbl_more2.setBounds(80, 6, 30, 15);
        lbl_more3.setBounds(80, 11, 30, 15);

        MainPanel.add(lbl_more1);
        MainPanel.add(lbl_more2);
        MainPanel.add(lbl_more3);

        JPanel panel = new JPanel();
        panel.setBackground(c3);
        panel.setBounds(30, 25, 120, 130);

        JButton btn_admin = new JButton("Admin Login");
        btn_admin.setBounds(10, 10, 100, 30);
        btn_admin.setBorder(new LineBorder(c2));
        btn_admin.setBackground(c1);
        btn_admin.setForeground(c3);
        btn_admin.addActionListener((ActionEvent e) -> {
            dispose();
            AdminLogin gp1 = new AdminLogin();
        });
        panel.add(btn_admin);

        lbl_more2.addMouseListener(new MouseListener() {
            @Override
            public void mouseClicked(MouseEvent e) {
                if(panel.isVisible()){
                    panel.setVisible(false);
                    MainPanel.remove(panel);
                    validate();
                    repaint();
                }
                else {
                    panel.setVisible(true);
                    MainPanel.add(panel);
                    validate();
                    repaint();
                }
            }

            @Override
            public void mousePressed(MouseEvent e) {
            }

            @Override
            public void mouseReleased(MouseEvent e) {
            }

            @Override
            public void mouseEntered(MouseEvent e) {
                panel.setVisible(true);
                MainPanel.add(panel);
                validate();
                repaint();
            }
            @Override
            public void mouseExited(MouseEvent e) {
            }
        });
        panel_one.add(MainPanel);

        addMouseListener(new MouseListener() {
            @Override
            public void mouseClicked(MouseEvent e) {
                panel.setVisible(false);
                MainPanel.remove(panel);
                validate();
                repaint();
            }

            @Override
            public void mousePressed(MouseEvent e) {}

            @Override
            public void mouseReleased(MouseEvent e) {}

            @Override
            public void mouseEntered(MouseEvent e) {}

            @Override
            public void mouseExited(MouseEvent e) {}
        });

        JLabel lbl_header = new JLabel("Speak Up");
        lbl_header.setFont (new Font ("TimesRoman", Font.BOLD, 35));
        lbl_header.setForeground(c2);
        lbl_header.setBounds(200, 70, 200, 60);
        panel_one.add(lbl_header);

        JLabel lbl_subHeader = new JLabel("Where everyone can speak!");
        lbl_subHeader.setFont(new Font("TimesRoman", Font.ITALIC, 18));
        lbl_subHeader.setForeground(c2);
        lbl_subHeader.setBounds(170, 130, 300, 25);
        panel_one.add(lbl_subHeader);

        JButton btn_loginAsUser = new JButton("Login As Public");
        btn_loginAsUser.setBorder(new LineBorder(c2));
        btn_loginAsUser.setBackground(c1);
        btn_loginAsUser.setForeground(c3);
        btn_loginAsUser.setBounds(215, 200, 150, 30);
        btn_loginAsUser.addActionListener((ActionEvent e) -> {
            dispose();
            LoginPage l1 = new LoginPage("Public");
        });
        panel_one.add(btn_loginAsUser);
```

```
        JButton btn_loginAsGovtServant = new JButton("Login As Government");
        btn_loginAsGovtServant.setBorder(new LineBorder(c2));
        btn_loginAsGovtServant.setBounds(190, 250, 200, 30);
        btn_loginAsGovtServant.setBackground(c1);
        btn_loginAsGovtServant.setForeground(c3);
        btn_loginAsGovtServant.addActionListener((ActionEvent e) -> {
            dispose();
            LoginPage l1 = new LoginPage("Government Official");
        });
        panel_one.add(btn_loginAsGovtServant);

        JLabel lbl_newUser = new JLabel("Don't have an account?");
        lbl_newUser.setBounds(220, 350, 200, 20);
        lbl_newUser.setFont(headerfont);
        panel_one.add(lbl_newUser);

        JButton btn_signUpAsPublic = new JButton("Sign Up As Public");
        btn_signUpAsPublic.setBorder(new LineBorder(c2));
        btn_signUpAsPublic.setBounds(200, 380, 170, 30);
        btn_signUpAsPublic.setBackground(c1);
        btn_signUpAsPublic.setForeground(c3);
        btn_signUpAsPublic.addActionListener((ActionEvent e) -> {
            dispose();
            RegistrationPublic p1 = new RegistrationPublic();
        });
        panel_one.add(btn_signUpAsPublic);

        add(panel_one);
        setResizable(false);
        setLayout(null);
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

## ADMIN MODULE →

## LOGIN PAGE:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.regex.Pattern;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPasswordField;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.UIManager;
import javax.swing.border.LineBorder;


public class AdminLogin extends JFrame
{
   JLabel lbl_phone, lbl_password, lbl_passErr, lbl_phoneErr;
   JTextField txt_phone;
   JPasswordField txt_password;

   AdminLogin()
   {
      setSize(570, 500);
      setTitle("Speak Up");
      setBounds(420, 100, 570, 500);

      Color c1=new Color(50, 105, 168);
      Color c2=new Color(0,0,0);
      Color c3=new Color(255, 255, 255);
      Color c4=new Color(93, 253, 146);

      getContentPane().setBackground(c3);
```

```
JLabel lbl_header = new JLabel("Admin Login");
lbl_header.setForeground(c2);
lbl_header.setBounds(200, 80, 300, 40);
Font headerFont = new Font("serif", Font.BOLD, 25);
lbl_header.setFont(headerFont);
add(lbl_header);

Font lblFont = new Font("Itc New Baskerville",Font.BOLD,13);

lbl_phone = new JLabel("Phone    :");
lbl_phone.setForeground(c2);
lbl_phone.setBounds(130, 150, 70, 30);
lbl_phone.setFont(lblFont);
add(lbl_phone);

lbl_phoneErr = new JLabel();
lbl_phoneErr.setFont(new Font("serif", Font.PLAIN, 12));
lbl_phoneErr.setBounds(250, 185, 150, 15);
lbl_phoneErr.setForeground(Color.red);

txt_phone = new JTextField();
txt_phone.setBounds(250, 150, 150, 30);
txt_phone.addKeyListener(new KeyListener() {
    @Override
    public void keyTyped(KeyEvent e) {

    }

    @Override
    public void keyPressed(KeyEvent e) {
    }

    @Override
    public void keyReleased(KeyEvent e) {
        String phone = txt_phone.getText();
        lbl_phoneErr.setText("Enter a valid phone number.");
        if(!Pattern.matches("^[6-9]{1}[0-9]{0,9}$", phone)){
            add(lbl_phoneErr);
            validate();
            repaint();
        }
        else{
            remove(lbl_phoneErr);
            validate();
            repaint();
        }
    }
});
add(txt_phone);
```

```
        lbl_password = new JLabel("Password  :");
        lbl_password.setForeground(c2);
        lbl_password.setBounds(130, 210, 120, 30);
        lbl_password.setFont(lblFont);
        add(lbl_password);

        lbl_passErr = new JLabel();
        lbl_passErr.setFont(new Font("serif", Font.PLAIN, 12));
        lbl_passErr.setBounds(250, 260, 150, 15);
        lbl_passErr.setForeground(Color.red);


        txt_password = new JPasswordField();
        txt_password.setBounds(250, 210, 150, 30);

        txt_password.addKeyListener(new KeyListener() {
            @Override
            public void keyTyped(KeyEvent e) {
            }

            @Override
            public void keyPressed(KeyEvent e) {
            }

            @Override
            public void keyReleased(KeyEvent e) {
                char[] password = txt_password.getPassword();
                if(password.length > 0){
                    remove(lbl_passErr);
                    validate();
                    repaint();
                }
            }
        });
        add(txt_password);

        JCheckBox check_password = new JCheckBox("Show Password");
        check_password.setForeground(c2);
        check_password.setBackground(c3);
        check_password.setBounds(250, 245, 150, 15);
        check_password.setFont(lblFont);
        check_password.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if(check_password.isSelected())
                    txt_password.setEchoChar('\u0000');
                else

txt_password.setEchoChar((Character)UIManager.get("PasswordField.echoChar"));
            }
} );
add(check_password);

JButton btn_submit = new JButton("Submit");
btn_submit.setBackground(c4);
btn_submit.setForeground(c2);
btn_submit.setBorder(new LineBorder(c2));
btn_submit.setBounds(280, 290, 100, 30);
btn_submit.addActionListener((ActionEvent e) -> {
   boolean flag = true;
   String phone = txt_phone.getText();
   char password[] = txt_password.getPassword();
   if(phone.length() < 1){
      lbl_phoneErr.setText("This field cannot be empty");
      add(lbl_phoneErr);
      validate();
      repaint();
      flag = false;
   } else if (!Pattern.matches("^[6-9]{1}[0-9]{9}$", phone)){
      lbl_phoneErr.setText("Enter a valid phone number");
      add(lbl_phoneErr);
      validate();
      repaint();
      flag = false;
   }
   if(password.length < 1){
      lbl_passErr.setText("This field cannot be empty");
      add(lbl_passErr);
      validate();
      repaint();
      flag = false;
   }
   if(flag) {
      verifyUser(phone, password);
   }
});
add(btn_submit);

JButton btn_goBack = new JButton("Back");
btn_goBack.setBackground(c1);
btn_goBack.setForeground(c3);
btn_goBack.setBounds(160, 290, 100, 30);

btn_goBack.addActionListener((ActionEvent e) -> {
   dispose();
   WelcomePage w1 = new WelcomePage();
});

add(btn_goBack);
```

```
        setResizable(false);
        setLayout(null);
        setVisible(true);
    }

    void verifyUser(String phone, char[] password){
        try{
            Connection con = DbConnection.getConnection();
            Statement stmt = con.createStatement();
            String query;
            query = "select * from admin where phone='"+phone+"'";
            ResultSet user = stmt.executeQuery(query);

            String hashedPassword = HashPassword.toHexString(new String(password));
            if(user.next()){
                String actualPassword = user.getString("password");
                if(actualPassword.equals(hashedPassword)){
                        AdminHomePage p1 = new AdminHomePage();
                        dispose();
                }
                else{
                    lbl_passErr.setText("Invalid Password");
                    add(lbl_passErr);
                    validate();
                    repaint();
                }
            }
            else{
                JTextArea lbl_noUser = new JTextArea("Account with given phone number does
not exist.");
                lbl_noUser.setWrapStyleWord(true);
                lbl_noUser.setLineWrap(true);
                lbl_noUser.setOpaque(false);
                lbl_noUser.setEditable(false);

                lbl_noUser.setBounds(210, 320, 180, 70);
                lbl_noUser.setFont(new Font("serif", Font.PLAIN, 15));
                lbl_noUser.setForeground(Color.red);
                add(lbl_noUser);
                validate();
                repaint();
            }
            con.close();
        } catch(SQLException ex) {
            System.out.println("exception occurred :" + ex);
        }
    }}
```

## HOME PAGE:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.border.LineBorder;

public class AdminHomePage extends JFrame{

    AdminHomePage()
    {
        setTitle("Speak Up");
        setSize(500, 500);
        setBounds(420, 100, 500, 500);

        Color c1=new Color(50, 105, 168);
        Color c2=new Color(0,0,0);
        Color c3=new Color(255, 255, 255);

        getContentPane().setBackground(c3);

        JPanel MainPanel = new JPanel();
        MainPanel.setBackground(c3);
        MainPanel.setBounds(330, 30, 150, 150);
        MainPanel.setLayout(null);
        JLabel lbl_more1 = new JLabel("___");
        JLabel lbl_more2 = new JLabel("___");
        JLabel lbl_more3 = new JLabel("___");
        lbl_more1.setBounds(80, 1, 30, 15);
        lbl_more2.setBounds(80, 6, 30, 15);
        lbl_more3.setBounds(80, 11, 30, 15);

        MainPanel.add(lbl_more1);
        MainPanel.add(lbl_more2);
        MainPanel.add(lbl_more3);

        JPanel panel = new JPanel();
        panel.setBackground(c3);
        panel.setBounds(50, 25, 80, 200);

        JButton btn_addAdmin = new JButton("Add Admin");
```

```
btn_addAdmin.setBorder(new LineBorder(c2));
btn_addAdmin.setBackground(c1);
btn_addAdmin.setForeground(c3);
btn_addAdmin.setBounds(190, 250, 100, 30);
btn_addAdmin.addActionListener(e -> {
    RegisterAdmin r1 = new RegisterAdmin();
});
panel.add(btn_addAdmin);


JButton btn_logout = new JButton("Logout");
btn_logout.setBorder(new LineBorder(c2));
btn_logout.setBackground(c1);
btn_logout.setForeground(c3);
btn_logout.setBounds(10, 150, 100, 20);
btn_logout.addActionListener(e -> {
    dispose();
    WelcomePage w1 = new WelcomePage();
});
panel.add(btn_logout);

lbl_more2.addMouseListener(new MouseListener() {
    @Override
    public void mouseClicked(MouseEvent e) {
        if(panel.isVisible()){
            panel.setVisible(false);
            MainPanel.remove(panel);
            validate();
            repaint();
        }
        else {
            panel.setVisible(true);
            MainPanel.add(panel);
            validate();
            repaint();
        }
    }

    @Override
    public void mousePressed(MouseEvent e) {
    }

    @Override
    public void mouseReleased(MouseEvent e) {
    }

    @Override
    public void mouseEntered(MouseEvent e) {
        panel.setVisible(true);
        MainPanel.add(panel);
        validate();
        repaint();
    }

    @Override
    public void mouseExited(MouseEvent e) {
    }
});
add(MainPanel);

addMouseListener(new MouseListener() {
    @Override
    public void mouseClicked(MouseEvent e) {
        panel.setVisible(false);
        MainPanel.remove(panel);
        validate();
        repaint();
    }

    @Override
    public void mousePressed(MouseEvent e) {}

    @Override
    public void mouseReleased(MouseEvent e) {}

    @Override
    public void mouseEntered(MouseEvent e) {}

    @Override
    public void mouseExited(MouseEvent e) {}
});

JLabel lbl_header = new JLabel("HI! ADMIN");
lbl_header.setBounds(180,100, 150, 30);
lbl_header.setForeground(c2);
lbl_header.setFont(new Font("serif", Font.BOLD, 25));
add(lbl_header);

JButton btn_registerGovt = new JButton("Register Government Official");
btn_registerGovt.setBorder(new LineBorder(c2));
btn_registerGovt.setBackground(c1);
btn_registerGovt.setForeground(c3);
btn_registerGovt.setBounds(120, 180, 250, 30);
btn_registerGovt.addActionListener(e -> {
    RegistrationGovtServant g1 = new RegistrationGovtServant();

});
add(btn_registerGovt);

JButton btn_viewOfficials = new JButton("View Governtment Officials");
```

```
        btn_viewOfficials.setBorder(new LineBorder(c2));
        btn_viewOfficials.setBackground(c1);
        btn_viewOfficials.setForeground(c3);
        btn_viewOfficials.setBounds(120, 230, 250, 30);
        btn_viewOfficials.addActionListener(e -> {
            AdminShowOfficials a3 = new AdminShowOfficials();
        });
        add(btn_viewOfficials);

        JButton btn_viewComplaints = new JButton("View Complaints");
        btn_viewComplaints.setBorder(new LineBorder(c2));
        btn_viewComplaints.setBackground(c1);
        btn_viewComplaints.setForeground(c3);
        btn_viewComplaints.setBounds(120, 280, 250, 30);
        btn_viewComplaints.addActionListener(e -> {
            AdminComplaints a2 = new AdminComplaints();
        });
        add(btn_viewComplaints);

        setResizable(false);
        setLayout(null);
        setVisible(true);
    }
}
```

# REGISTER ADMIN:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.awt.HeadlessException;
import java.awt.event.ActionEvent;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.regex.Pattern;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPasswordField;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.UIManager;
import javax.swing.border.LineBorder;


public class RegisterAdmin extends JFrame{
   JLabel lbl_phone, lbl_password, lbl_passErr, lbl_phoneErr;
   JTextField txt_phone;
   JPasswordField txt_password;
   JTextArea lbl_knownUser = new JTextArea();

   Color c1=new Color(50, 105, 168);
   Color c2=new Color(0,0,0);
   Color c3=new Color(255, 255, 255);
   Color c4 = new Color(93, 253, 146);

   RegisterAdmin()
   {
      setSize(570, 500);
      setTitle("Speak Up");
      setBounds(400, 100, 570, 500);

      getContentPane().setBackground(c3);

      JLabel lbl_header = new JLabel("REGISTER ADMIN");
      lbl_header.setForeground(c2);
      lbl_header.setBounds(155, 80, 300, 40);
```

```
Font headerFont = new Font("serif", Font.BOLD, 25);
lbl_header.setFont(headerFont);
add(lbl_header);

Font lblFont = new Font("Itc New Baskerville",Font.BOLD,13);

lbl_phone = new JLabel("Phone Number  :");
lbl_header.setForeground(c2);
lbl_phone.setBounds(100, 150, 200, 30);
lbl_phone.setFont(lblFont);
add(lbl_phone);

lbl_phoneErr = new JLabel();
lbl_phoneErr.setFont(new Font("serif", Font.PLAIN, 12));
lbl_phoneErr.setBounds(250, 185, 150, 15);
lbl_phoneErr.setForeground(Color.red);

txt_phone = new JTextField();
txt_phone.setBorder(new LineBorder(c2));
txt_phone.setBounds(250, 150, 150, 30);
txt_phone.addKeyListener(new KeyListener() {
   @Override
   public void keyTyped(KeyEvent e) {

   }

   @Override
   public void keyPressed(KeyEvent e) {
   }

   @Override
   public void keyReleased(KeyEvent e) {
      String phone = txt_phone.getText();
      lbl_phoneErr.setText("Enter a valid phone number.");
      if(!Pattern.matches("^[6-9]{1}[0-9]{0,9}$", phone)){
         add(lbl_phoneErr);
         validate();
         repaint();
      }
      else{
         remove(lbl_phoneErr);
         validate();
         repaint();
      }
   }
});
add(txt_phone);

lbl_password = new JLabel("Password  :");
lbl_header.setForeground(c2);
lbl_password.setBounds(100, 210, 120, 30);
lbl_password.setFont(lblFont);
add(lbl_password);

lbl_passErr = new JLabel();
lbl_passErr.setFont(new Font("serif", Font.PLAIN, 12));
lbl_passErr.setBounds(250, 260, 150, 15);
lbl_passErr.setForeground(Color.red);

txt_password = new JPasswordField();
txt_password.setBorder(new LineBorder(c2));
txt_password.setBounds(250, 210, 150, 30);

txt_password.addKeyListener(new KeyListener() {
  @Override
  public void keyTyped(KeyEvent e) {
  }

  @Override
  public void keyPressed(KeyEvent e) {
  }

  @Override
  public void keyReleased(KeyEvent e) {
    char[] password = txt_password.getPassword();
    if(password.length > 0){
      remove(lbl_passErr);
      validate();
      repaint();
    }
  }
});
add(txt_password);

JCheckBox check_password = new JCheckBox("Show Password");
check_password.setBackground(c3);
check_password.setForeground(c2);
check_password.setBounds(250, 245, 150, 15);
check_password.setFont(lblFont);
check_password.addActionListener((ActionEvent e) -> {
  if(check_password.isSelected())
    txt_password.setEchoChar('\u0000');
  else
    txt_password.setEchoChar((Character)UIManager.get("PasswordField.echoChar"));
});
add(check_password);

JButton btn_submit = new JButton("Submit");
btn_submit.setBackground(c4);
btn_submit.setForeground(c2);
btn_submit.setBorder(new LineBorder(c2));
btn_submit.setBounds(280, 290, 100, 30);
```

```
    btn_submit.addActionListener((ActionEvent e) -> {
        boolean flag = true;
        String phone = txt_phone.getText();
        char password[] = txt_password.getPassword();
        if(phone.length() < 1){
            lbl_phoneErr.setText("This field cannot be empty");
            add(lbl_phoneErr);
            validate();
            repaint();
            flag = false;
        } else if (!Pattern.matches("^[6-9]{1}[0-9]{9}$", phone)){
            lbl_phoneErr.setText("Enter a valid phone number");
            add(lbl_phoneErr);
            validate();
            repaint();
            flag = false;
        }
        if(password.length < 1){
            lbl_passErr.setText("This field cannot be empty");
            add(lbl_passErr);
            validate();
            repaint();
            flag = false;
        }
        if(flag) {
            verifyUser(phone, password);
        }
    });
    add(btn_submit);

    JButton btn_goBack = new JButton("Back");
    btn_goBack.setBorder(new LineBorder(c2));
    btn_goBack.setBackground(c1);
    btn_goBack.setForeground(c3);
    btn_goBack.setBounds(160, 290, 100, 30);

    btn_goBack.addActionListener((ActionEvent e) -> {
        dispose();
    });

    add(btn_goBack);

    setResizable(false);
    setLayout(null);
    setVisible(true);
}

void verifyUser(String phone, char[] password){
    try{
        Connection con = DbConnection.getConnection();
        PreparedStatement stmt;
```

```
      stmt = con.prepareStatement("select * from admin where phone=(?)");
      stmt.setString(1, phone);
      ResultSet user = stmt.executeQuery();
      if(user.next()){
        remove(lbl_knownUser);
        lbl_knownUser.setWrapStyleWord(true);
        lbl_knownUser.setLineWrap(true);
        lbl_knownUser.setOpaque(false);
        lbl_knownUser.setEditable(false);

        lbl_knownUser.setBounds(150, 570, 250, 70);
        lbl_knownUser.setFont(new Font("serif", Font.PLAIN, 15));
        lbl_knownUser.setForeground(Color.red);

        lbl_knownUser.setText("User with the given phone, email and unique id already exists.");
        add(lbl_knownUser);
        validate();
        repaint();
      }
      else {
        Statement addStmt = con.createStatement();
        String hashedPassword = HashPassword.toHexString(new String(password));
        String query = "insert into admin(phone, password)
values('"+phone+"','"+hashedPassword+"')";
        int result = addStmt.executeUpdate(query);
        System.out.println("Result :" + result);
        UIManager UI=new UIManager();
        UI.put("OptionPane.background", c3);
        UI.put("Panel.background", c3);
        JOptionPane.showMessageDialog(this, "Account sucessfully created!");
        dispose();
      }
      con.close();

    } catch(HeadlessException | SQLException ex) {
      System.out.println("exception occurred :" + ex);
    }
  }}
```

# REGISTER GOVERNMENT OFFICIALS:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.awt.HeadlessException;
import java.awt.event.ActionEvent;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Arrays;
import java.util.regex.Pattern;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPasswordField;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.UIManager;
import javax.swing.border.LineBorder;

public class RegistrationGovtServant extends JFrame{

   JTextField txt_name,txt_phn_num,txt_email,txt_uqu_id;
   JPasswordField pas_pass,pas_repass;
   JLabel lbl_name_err, lbl_email_err, lbl_phn_err, lbl_uqid_err, lbl_city_err, lbl_pass_err,
lbl_repass_err;
   JComboBox cb_sector, cb_city;
   JLabel l_title,  lbl_name,lbl_phn_num,lbl_sector,lbl_email,lbl_uqu_id,lbl_pass,lbl_repass,lbl_city;
   JButton btn_reg,btn_back;
   JTextArea lbl_knownUser = new JTextArea();
   boolean flag;

   Color c1=new Color(50, 105, 168);
   Color c2=new Color(0,0,0);
   Color c3=new Color(255, 255, 255);
   Color c4 = new Color (93, 253, 146);

   public RegistrationGovtServant()
   {
      setTitle("Speak Up");
      setBounds(350, 70, 600, 500);
```

```
getContentPane().setBackground(c3);

Font headerfont = new Font("Itc New Baskerville",Font.BOLD,13);

l_title = new JLabel("GOVERNMENT OFFICIAL REGISTRATION");
l_title.setForeground(c2);
l_title.setFont(new Font("Serif", Font.BOLD, 20));
l_title.setBounds(80, 30, 500, 30);
add(l_title);

lbl_name = new JLabel("Name  : ");
lbl_name.setForeground(c2);
lbl_name.setFont(headerfont);
lbl_name.setBounds(100,83,150,20);
add(lbl_name);

txt_name = new JTextField();
txt_name.setBorder(new LineBorder(c2));
txt_name.setBounds(280,80,200,30);
txt_name.addKeyListener(new KeyListener() {
   @Override
   public void keyTyped(KeyEvent e) {
   }

   @Override
   public void keyPressed(KeyEvent e) {
   }

   @Override
   public void keyReleased(KeyEvent e) {
      String name = txt_name.getText();
      if(name.length() < 1){
         lbl_name_err.setText("This Field cannot be empty.");
         add(lbl_name_err);
      } else {
         remove(lbl_name_err);
      }
      validate();
      repaint();
   }
});
add(txt_name);

lbl_name_err = new JLabel();
lbl_name_err.setBounds(280, 111, 150, 15);
lbl_name_err.setFont(new Font("serif", Font.PLAIN, 12));
lbl_name_err.setForeground(Color.red);

lbl_phn_num = new JLabel("Phone Number  : ");
lbl_phn_num.setFont(headerfont);
lbl_phn_num.setBounds(100,133,150,20);
add(lbl_phn_num);

txt_phn_num = new JTextField();
txt_phn_num.setBorder(new LineBorder(c2));
txt_phn_num.setBounds(280,130,200,30);
txt_phn_num.addKeyListener(new KeyListener() {
   @Override
   public void keyTyped(KeyEvent e) {

   }

   @Override
   public void keyPressed(KeyEvent e) {
   }

   @Override
   public void keyReleased(KeyEvent e) {
      String phone = txt_phn_num.getText();
      lbl_phn_err.setText("Enter a valid phone number.");
      if(!Pattern.matches("^[6-9]{1}[0-9]{0,9}$", phone)){
         add(lbl_phn_err);
         validate();
         repaint();
      }
      else{
         remove(lbl_phn_err);
         validate();
         repaint();
      }
   }
});

add(txt_phn_num);

lbl_phn_err = new JLabel();
lbl_phn_err.setBounds(280, 161,150, 15);
lbl_phn_err.setFont(new Font("serif", Font.PLAIN, 12));
lbl_phn_err.setForeground(Color.red);

lbl_email = new JLabel("Email-ID  : ");
lbl_email.setForeground(c2);
lbl_email.setFont(headerfont);
lbl_email.setBounds(100,183,100,20);
add(lbl_email);

txt_email = new JTextField();
txt_email.setBorder(new LineBorder(c2));
txt_email.setBounds(280,180,200,30);
txt_email.addKeyListener(new KeyListener() {
   @Override
```

```
    public void keyTyped(KeyEvent e) {
    }

    @Override
    public void keyPressed(KeyEvent e) {
    }

    @Override
    public void keyReleased(KeyEvent e) {
        String email = txt_email.getText();
        if(email.length() < 1){
            lbl_email_err.setText("This Field cannot be empty.");
            add(lbl_email_err);
        } else {
            remove(lbl_email_err);
        }
        validate();
        repaint();
    }
});
add(txt_email);

lbl_email_err = new JLabel();
lbl_email_err.setBounds(280, 211, 150, 15);
lbl_email_err.setFont(new Font("serif", Font.PLAIN, 12));
lbl_email_err.setForeground(Color.red);


lbl_sector = new JLabel("Sector  : ");
lbl_sector.setForeground(c2);
lbl_sector.setFont(headerfont);
lbl_sector.setBounds(100,233,100,20);
add(lbl_sector);

String sector[]={"Corporation" ,"Electricity", "Water"};
cb_sector = new JComboBox(sector);
cb_sector.setBorder(new LineBorder(c2));
cb_sector.setBackground(c3);
cb_sector.setBounds(280,230,200,30);
add(cb_sector);

lbl_city = new JLabel("City  : ");
lbl_city.setForeground(c2);
lbl_city.setFont(headerfont);
lbl_city.setBounds(100,283,100,20);
add(lbl_city);

String cities[] = {"City 1", "City 2"};
cb_city = new JComboBox(cities);
cb_city.setBorder(new LineBorder(c2));
cb_city.setBackground(c3);
cb_city.setBounds(280,280,200,30);
add(cb_city);

lbl_uqu_id = new JLabel("Aadhar Number  : ");
lbl_sector.setForeground(c2);
lbl_uqu_id.setFont(headerfont);
lbl_uqu_id.setBounds(100,333,200,20);
add(lbl_uqu_id);

txt_uqu_id = new JTextField();
txt_uqu_id.setBorder(new LineBorder(c2));
txt_uqu_id.setBounds(280,330,200,30);
txt_uqu_id.addKeyListener(new KeyListener() {
   @Override
   public void keyTyped(KeyEvent e) {
   }

   @Override
   public void keyPressed(KeyEvent e) {
   }

   @Override
   public void keyReleased(KeyEvent e) {
      String uqid = txt_uqu_id.getText();
      if(uqid.length() < 1){
         lbl_uqid_err.setText("This Field cannot be empty.");
         add(lbl_uqid_err);
      } else if(uqid.length() > 12) {
         lbl_uqid_err.setText("Enter a valid Aadhaar");
         add(lbl_uqid_err);
      } else {
         remove(lbl_uqid_err);
      }
      validate();
      repaint();
   }
});
add(txt_uqu_id);

lbl_uqid_err = new JLabel();
lbl_uqid_err.setBounds(280, 361, 150, 15);
lbl_uqid_err.setFont(new Font("serif", Font.PLAIN, 12));
lbl_uqid_err.setForeground(Color.red);

lbl_pass = new JLabel("Password  : ");
lbl_pass.setForeground(c2);
lbl_pass.setFont(headerfont);
lbl_pass.setBounds(100,383,170,20);
add(lbl_pass);

pas_pass = new JPasswordField();
```

```
pas_pass.setBorder(new LineBorder(c2));
pas_pass.setBounds(280,380,200,30);
pas_pass.addKeyListener(new KeyListener() {
    @Override
    public void keyTyped(KeyEvent e) {
    }

    @Override
    public void keyPressed(KeyEvent e) {
    }

    @Override
    public void keyReleased(KeyEvent e) {
        String pass = new String(pas_pass.getPassword());
        if(pass.length() < 1){
            lbl_pass_err.setText("This Field cannot be empty.");
            add(lbl_pass_err);
        } else {
            remove(lbl_pass_err);
        }
        validate();
        repaint();
    }
});
add(pas_pass);

lbl_pass_err = new JLabel();
lbl_pass_err.setBounds(280,431, 200, 15);
lbl_pass_err.setFont(new Font("serif", Font.PLAIN, 11));
lbl_pass_err.setForeground(Color.red);

JCheckBox check_password = new JCheckBox("Show Password");
check_password.setBackground(c3);
check_password.setForeground(c2);
check_password.setBounds(280, 415, 150, 15);
check_password.setFont(new Font("serif", Font.PLAIN, 13));
check_password.addActionListener((ActionEvent e) -> {
    if(check_password.isSelected())
        pas_pass.setEchoChar('\u0000');
    else
        pas_pass.setEchoChar((Character)UIManager.get("PasswordField.echoChar"));
});
add(check_password);

lbl_repass = new JLabel("Confirm Password  : ");
lbl_repass.setForeground(c2);
lbl_repass.setFont(headerfont);
lbl_repass.setBounds(100,453,170,20);
add(lbl_repass);

pas_repass = new JPasswordField();
```

```
pas_repass.setBorder(new LineBorder(c2));
pas_repass.setBounds(280,450,200,30);
pas_repass.addKeyListener(new KeyListener() {
   @Override
   public void keyTyped(KeyEvent e) {
   }

   @Override
   public void keyPressed(KeyEvent e) {
   }

   @Override
   public void keyReleased(KeyEvent e) {
      String repass = new String(pas_repass.getPassword());
      if(repass.length() < 1){
         lbl_repass_err.setText("This Field cannot be empty.");
         add(lbl_repass_err);
      } else {
         remove(lbl_repass_err);
      }
      validate();
      repaint();
   }
});
add(pas_repass);

lbl_repass_err = new JLabel();
lbl_repass_err.setBounds(280, 501, 150, 15);
lbl_repass_err.setFont(new Font("serif", Font.PLAIN, 11));
lbl_repass_err.setForeground(Color.red);

JCheckBox check_re_password = new JCheckBox("Show Password");
check_re_password.setBackground(c3);
check_re_password.setForeground(c2);
check_re_password.setBounds(280, 485, 150, 15);
check_re_password.setFont(new Font("serif", Font.PLAIN, 13));
check_re_password.addActionListener((ActionEvent e) -> {
   if(check_re_password.isSelected())
      pas_repass.setEchoChar('\u0000');
   else
      pas_repass.setEchoChar((Character)UIManager.get("PasswordField.echoChar"));
});
add(check_re_password);

btn_reg = new JButton("Register");
btn_reg.setBackground(c4);
btn_reg.setForeground(c2);
btn_reg.setBorder(new LineBorder(c2));
btn_reg.setBounds(320,520,100,30);
btn_reg.addActionListener((ActionEvent e) -> {
   handleSubmit();
```

```
        });

        add(btn_reg);

        btn_back = new JButton("Back");
        btn_back.setBackground(c1);
        btn_back.setForeground(c3);
        btn_back.setBounds(150,520,100,30);
        btn_back.addActionListener(e -> {
            dispose();
        });

        add(btn_back);

        setSize(600, 650);
        setResizable(false);
        setLayout(null);
        setVisible(true);
    }

    void handleSubmit() {
        String name = txt_name.getText();
        String phone = txt_phn_num.getText();
        String email = txt_email.getText();
        char[] password = pas_pass.getPassword();
        char[] confirmPassword = pas_repass.getPassword();
        String unique_id = txt_uqu_id.getText();
        String city = cb_city.getSelectedItem().toString();

        boolean flag = CheckValidGovt.checkValid(this, name, email, city, phone, lbl_name_err,
lbl_email_err, lbl_city_err, lbl_phn_err);

        if(password.length < 1){
            lbl_pass_err.setText("This Field cannot be empty.");
            add(lbl_pass_err);
            flag = false;
        } else if (!Pattern.matches("^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z]).{8,32}$", new String(password))){
            pas_pass.setText("");
            pas_repass.setText("");
            lbl_pass_err.setText("Password must follow the provided rules");
            add(lbl_pass_err);
            flag = false;
        } else if (!Arrays.equals(password,confirmPassword)) {
            lbl_pass_err.setText("The passwords must match.");
            add(lbl_pass_err);
            lbl_repass_err.setText("The passwords must match.");
            add(lbl_repass_err);
            pas_pass.setText("");
            pas_repass.setText("");
            flag = false;
        }
        if(confirmPassword.length < 1){
            lbl_repass_err.setText("This Field cannot be empty.");
            add(lbl_repass_err);
            flag = false;
        }
        if(unique_id.length() < 1){
            lbl_uqid_err.setText("This Field cannot be empty.");
            add(lbl_uqid_err);
            flag = false;
        } else if(unique_id.length() != 12) {
            lbl_uqid_err.setText("Enter a valid Aadhaar.");
            add(lbl_uqid_err);
            flag = false;
        }

        if(flag){
            addUser(name, phone, email, password, unique_id, city);
        }
        validate();
        repaint();
    }
    void addUser(String name, String phone, String email, char[] password, String unique_id, String
city ){
        String sector = cb_sector.getSelectedItem().toString();
        try{
            Connection con = DbConnection.getConnection();
            PreparedStatement stmt;
            stmt = con.prepareStatement("select * from govt_servant where phone=(?) or email=(?) or
unique_id=(?)");
            stmt.setString(1, phone);
            stmt.setString(2, email);
            stmt.setString(3, unique_id);
            ResultSet user = stmt.executeQuery();
            if(user.next()){
              remove(lbl_knownUser);
              lbl_knownUser.setWrapStyleWord(true);
              lbl_knownUser.setLineWrap(true);
              lbl_knownUser.setOpaque(false);
              lbl_knownUser.setEditable(false);

              lbl_knownUser.setBounds(150, 570, 250, 70);
              lbl_knownUser.setFont(new Font("serif", Font.PLAIN, 15));
              lbl_knownUser.setForeground(Color.red);

              if(user.getString("phone").equals(phone) && user.getString("email").equals(email) &&
user.getString("unique_id").equals(unique_id)) {
          lbl_knownUser.setText("User with the given phone, email and unique id already exists.");
              } else if (user.getString("phone").equals(phone)){
          lbl_knownUser.setText("User with the given phone already exists.");
              } else if (user.getString("email").equals(email)) {
          lbl_knownUser.setText("User with the given email already exists.");
              } else if (user.getString("unique_id").equals(unique_id)) {
          lbl_knownUser.setText("User with the given unique id already exists.");
              }
              add(lbl_knownUser);
              validate();
              repaint();
            }
            else {
              Statement addStmt = con.createStatement();
              String hashedPassword = HashPassword.toHexString(new String(password));
              String query = "insert into govt_servant(name, phone, email, sector, password, city,
unique_id) values('"+name+"', '"+phone+"', '"+email+"', '"+sector+"', '"+hashedPassword+"',
'"+city+"', '"+unique_id+"')";
              int result = addStmt.executeUpdate(query);
              System.out.println("Result :" + result);
              JOptionPane.showMessageDialog(this, "Account sucessfully created!");
              dispose();
            }
            con.close();
        } catch(HeadlessException | SQLException ex) {
            System.out.println("exception occurred :" + ex);
        }
    }}
```

## VIEW GOVERNMENT OFFICIALS:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Vector;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.border.LineBorder;

public class AdminShowOfficials extends JFrame {
    JPanel panel;
    JTable table;
    JScrollPane scrl;
    JComboBox cb_city, cb_sector;

    Color c1=new Color(50, 105, 168);
    Color c2=new Color(0,0,0);
    Color c3=new Color(255, 255, 255);

    AdminShowOfficials()
    {
        setSize(570, 400);
        setBounds(385, 140, 570, 400);
        setTitle("Speak Up");

        panel = new JPanel();
        panel.setLayout(null);
        panel.setBackground(c3);

        JLabel lbl_head = new JLabel("Government Officials");
        lbl_head.setBounds(150, 20, 250, 30);
        lbl_head.setFont(new Font("serif", Font.BOLD, 22));
        panel.add(lbl_head);

        String[] sectors = {"--Select Sector--","Corporation", "Electricity", "Water"};
        cb_sector = new JComboBox(sectors);
        cb_sector.setBorder(new LineBorder(c2));
        cb_sector.setBackground(c3);
        cb_sector.setForeground(c2);
        cb_sector.setBounds(70, 80, 150, 30);
        cb_sector.addActionListener( e -> {
            String sector = cb_sector.getSelectedItem().toString();
            try{
              String query1;
              if(cb_sector.getSelectedIndex() >= 1){
                if(cb_city.getSelectedIndex() == 0)
                  query1 = "select * from govt_servant where sector='" + sector + "'";
                else{
                  String city = cb_city.getSelectedItem().toString();
                  query1 = "select * from govt_servant where sector='"+sector+"' and
    city='"+city+"'";

                }
              }
              else{
                if(cb_city.getSelectedIndex() == 0)
                  query1 = "select * from govt_servant";
                else{
                  String city = cb_city.getSelectedItem().toString();
                  query1 = "select * from govt_servant where city='"+city+"'";
                }
              }
              Connection con = DbConnection.getConnection();
              Statement stmt1 = con.createStatement();
              ResultSet rs1 = stmt1.executeQuery(query1);
              panel.remove(scrl);
              showTable(rs1);
            } catch(SQLException ex){
              System.out.println("Exception occurred");
            }
        });

        String[] cities = {"--Select City--","City 1", "City 2"};

        cb_city = new JComboBox(cities);
        cb_city.setBorder(new LineBorder(c2));
        cb_city.setBackground(c3);
        cb_city.setForeground(c2);
        cb_city.setBounds(320, 80, 150, 30);
        cb_city.addActionListener( e -> {
          String city = cb_city.getSelectedItem().toString();
          try{
            String query1;
            if(cb_city.getSelectedIndex() >= 1){
              if(cb_sector.getSelectedIndex() == 0)
                query1 = "select * from govt_servant where city='" + city + "'";
              else{
```

```
                        String sector = cb_sector.getSelectedItem().toString();
                        query1 = "select * from govt_servant where sector='"+sector+"' and
city='"+city+"'";
                    }
                }
                else{
                    if(cb_sector.getSelectedIndex() == 0)
                        query1 = "select * from govt_servant";
                    else{
                        String sector = cb_sector.getSelectedItem().toString();
                        query1 = "select * from govt_servant where sector='"+sector+"'";
                    }
                }
                Connection con = DbConnection.getConnection();
                Statement stmt1 = con.createStatement();
                ResultSet rs1 = stmt1.executeQuery(query1);
                panel.remove(scrl);
                showTable(rs1);
            } catch(SQLException ex){
                System.out.println("Exception occurred");
            }
        });

        panel.add(cb_sector);
        panel.add(cb_city);

        try {
            Connection con = DbConnection.getConnection();
            con = DbConnection.getConnection();
            Statement stmt = con.createStatement();
            String query = "select * from govt_servant";
            ResultSet rs = stmt.executeQuery(query);
            showTable(rs);
        } catch(SQLException ex) {
            System.out.println("Exception occurred  "+ex);
        }

        JButton btn_back = new JButton("Back");
        btn_back.setBorder(new LineBorder(c2));
        btn_back.setBackground(c1);
        btn_back.setForeground(c3);
        btn_back.setBounds(220, 320, 100, 30);
        btn_back.addActionListener(e -> {
            dispose();
        });
        add(btn_back);

        add(panel);

        setResizable(false);
        setVisible(true);
        setLayout(null);


    }
    void showTable(ResultSet rs) {

        Vector<String> col = new Vector<>();
        col.add("Name");
        col.add("Phone");
        col.add("Sector");
        col.add("City");

        Vector<Vector> arr = new Vector<>();
        try {
            while(rs.next()) {
                Vector<String> list = new Vector<>();
                list.add(rs.getString("name"));
                list.add(rs.getString("phone"));
                list.add(rs.getString("sector"));
                list.add(rs.getString("city"));
                arr.add(list);
            }
        } catch(SQLException ex) {
            System.out.println("Exception occurred "+ex);
        }

        table = new JTable(arr, col);
        table.setBackground(c3);
        table.setEnabled(false);
        scrl = new JScrollPane(table);
        scrl.setBounds(20, 150, 500, 110);
        panel.add(scrl);
        panel.setBackground(c3);

        validate();
        repaint();

    }
}
```

## VIEW COMPLAINTS:

```
package speak_up;

import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.border.LineBorder;

public class AdminComplaints extends JFrame {

    ResultSet rs;
    AdminComplaints() {
        setSize(600, 600);
        setTitle("Speak Up");
        setBounds(420, 100, 550, 500);

        Color c1=new Color(50, 105, 168);
        Color c2=new Color(0,0,0);
        Color c3=new Color(255, 255, 255);

        getContentPane().setBackground(c3);

        try {
            Connection con = DbConnection.getConnection();
            String query1 = "select * from complaints";
            Statement state = con.createStatement();
            rs = state.executeQuery(query1);
            con.close();
        } catch(SQLException exp){
            JOptionPane.showMessageDialog(this,"Error Occured while connecting to
database...");
        }

        JLabel main_lbl = new JLabel("PUBLIC COMPLAINTS");
        main_lbl.setForeground(c2);
        main_lbl.setFont(new Font("Verdana", Font.BOLD, 18));
        main_lbl.setBounds(150, 10, 300, 50);
```

```
add(main_lbl);
setLayout(new FlowLayout());
try{

    if(rs.next()){
        JPanel panel_mainPanel = new JPanel();
        panel_mainPanel.setBackground(c3);
        panel_mainPanel.setLayout(new GridLayout(10, 1, 25, 2));
        panel_mainPanel.setBounds(25, 50, 500, 300);

        GridLayout grid = new GridLayout(1, 5, 20, 20);

        Font headingFont = new Font("serif", Font.BOLD, 17);

        JPanel panel_header = new JPanel(grid);
        panel_header.setBackground(c3);

        JLabel lbl_headName = new JLabel("Sector");
        lbl_headName.setForeground(c2);
        lbl_headName.setFont(headingFont);

        JLabel lbl_headPostedAt = new JLabel("Posted At");
        lbl_headPostedAt.setForeground(c2);
        lbl_headPostedAt.setFont(headingFont);

        JLabel lbl_headStatus = new JLabel("Status");
        lbl_headStatus.setForeground(c2);
        lbl_headStatus.setFont(headingFont);


        JLabel lbl_headArea = new JLabel("City");
        lbl_headArea.setForeground(c2);
        lbl_headArea.setFont(headingFont);

        JLabel lbl_headView = new JLabel("View");
        lbl_headArea.setForeground(c2);
        lbl_headArea.setFont(headingFont);

        panel_header.add(lbl_headName);
        panel_header.add(lbl_headPostedAt);
        panel_header.add(lbl_headStatus);
        panel_header.add(lbl_headArea);
        panel_header.add(lbl_headView);

        panel_mainPanel.add(panel_header);
        do {
            JPanel panel = new JPanel(grid);
            panel.setBackground(c3);

            JLabel lbl_name = new JLabel(rs.getString("sector"));
            JLabel lbl_sector = new JLabel(rs.getString("posted_at"));
            JLabel lbl_status = new JLabel(rs.getString("status").toUpperCase());
            JLabel lbl_area = new JLabel(rs.getString("city"));

            JButton btn_view = new JButton("View");
            btn_view.setBorder(new LineBorder(c2));
            btn_view.setBackground(c1);
            btn_view.setForeground(c3);
            int id = rs.getInt("Id");
            btn_view.addActionListener(e -> {
                  showComplaint(id);
            });

            panel.add(lbl_name);
            panel.add(lbl_sector);
            panel.add(lbl_status);
            panel.add(lbl_area);
            panel.add(btn_view);
            panel_mainPanel.add(panel);
         } while(rs.next());
          add(panel_mainPanel);

      }
      else {
         JLabel lbl_empty = new JLabel("There are no complaints.");
         lbl_empty.setFont(new Font("serif", Font.PLAIN, 17));
         lbl_empty.setBounds(180, 150, 200, 100);
         add(lbl_empty);
      }
   } catch(SQLException ex) {
      System.out.println("Exception while showing complaints. "+ex);
   }

   JButton btn_back = new JButton("Back");
   btn_back.setBorder(new LineBorder(c2));
   btn_back.setBackground(c1);
   btn_back.setForeground(c3);
   btn_back.setBounds(220, 400, 100, 30);
   btn_back.addActionListener(e -> {
      dispose();
   });
   add(btn_back);

   setResizable(false);
   setLayout(null);
   setVisible(true);

}
void showComplaint(int id) {
   try {
```

```
            Connection con = DbConnection.getConnection();
            Statement stmt = con.createStatement();
            String query = "select c.id, p.name, p.phone, g.name as gname, g.phone as gphone,
c.city, c.sector, c.description, c.status, c.area, c.posted_at from complaints c join public p on
c.person_id=p.id join govt_servant g on g.city=c.city where c.person_id=p.id and
g.city=c.city and g.sector=c.sector and c.id="+id;
            ResultSet result = stmt.executeQuery(query);
            result.next();
            AdminShowComplaint a1 = new AdminShowComplaint(result);
        } catch(SQLException ex) {
            System.out.println("Exeption occurred while showing complaints..");
        }
    }

}
```

## SHOW COMPLAINTS:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.awt.GridLayout;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.border.LineBorder;

public class AdminShowComplaint extends JFrame {

    public AdminShowComplaint(ResultSet rs){
        setTitle("Speak Up");
        setSize(500, 600);
        setBounds(400, 100, 500, 600);

        Color c1=new Color(50, 105, 168);
        Color c2=new Color(0,0,0);
        Color c3=new Color(255, 255, 255);
        Color c4 =new Color(255, 0, 0);
```

```
getContentPane().setBackground(c3);

try {
   int id = rs.getInt("id");

   Font keyFont = new Font("serif", Font.BOLD, 18);
   Font valueFont = new Font("serif", Font.PLAIN, 18);

   JLabel lbl_head = new JLabel("COMPLAINT");
   lbl_head.setForeground(c2);
   lbl_head.setBounds(170, 20, 200, 30);
   lbl_head.setFont(new Font("serif", Font.BOLD, 23));
   add(lbl_head);

   JPanel mainPanel = new JPanel(new GridLayout(8, 2));
   mainPanel.setBackground(c3);
   mainPanel.setBounds(20, 50, 440, 550);
   mainPanel.setLayout(null);

   JLabel lbl_name = new JLabel("Complaint By :");
   lbl_name.setForeground(c2);
   lbl_name.setFont(keyFont);
   lbl_name.setBounds(40, 20, 150, 30);
   mainPanel.add(lbl_name);

   JLabel lbl_name_state = new JLabel(rs.getString("name"));
   lbl_name_state.setBounds(220, 20, 100, 30);
   lbl_name_state.setFont(valueFont);
   mainPanel.add(lbl_name_state);

   JLabel lbl_phone = new JLabel("Contact No. :");
   lbl_phone.setForeground(c2);
   lbl_phone.setBounds(40, 60, 200, 30);
   lbl_phone.setFont(keyFont);
   mainPanel.add(lbl_phone);

   JLabel lbl_phone_state = new JLabel(rs.getString("phone"));
   lbl_phone_state.setBounds(220, 60, 100, 30);
   lbl_phone_state.setFont(valueFont);
   mainPanel.add(lbl_phone_state);

   JLabel lbl_sector = new JLabel("Sector:");
   lbl_sector.setForeground(c2);
   lbl_sector.setBounds(40, 100, 150, 30);
   lbl_sector.setFont(keyFont);
   mainPanel.add(lbl_sector);
```

```
JLabel lbl_sector_state = new JLabel(rs.getString("sector"));
lbl_sector_state.setBounds(220, 100, 100, 30);
lbl_sector_state.setFont(valueFont);
mainPanel.add(lbl_sector_state);

JLabel lbl_official = new JLabel("Official :");
lbl_official.setForeground(c2);
lbl_official.setBounds(40, 140, 150, 30);
lbl_official.setFont(keyFont);
mainPanel.add(lbl_official);

JLabel lbl_official_state = new JLabel(rs.getString("gname"));
lbl_official_state.setBounds(220, 140, 100, 30);
lbl_official_state.setFont(valueFont);
mainPanel.add(lbl_official_state);

JLabel lbl_offl_phone = new JLabel("Official Contact No. :");
lbl_offl_phone.setForeground(c2);
lbl_offl_phone.setBounds(40, 180, 180, 30);
lbl_offl_phone.setFont(keyFont);
mainPanel.add(lbl_offl_phone);

JLabel lbl_offl_phone_state = new JLabel(rs.getString("gphone"));
lbl_offl_phone_state.setBounds(220, 180, 100, 30);
lbl_offl_phone_state.setFont(valueFont);
mainPanel.add(lbl_offl_phone_state);


JLabel lbl_city = new JLabel("City :");
lbl_city.setForeground(c2);
lbl_city.setBounds(40, 220, 150, 30);
lbl_city.setFont(keyFont);
mainPanel.add(lbl_city);

JLabel lbl_city_state = new JLabel(rs.getString("city"));
lbl_city_state.setBounds(220, 220, 100, 30);
lbl_city_state.setFont(valueFont);
mainPanel.add(lbl_city_state);

JLabel lbl_area = new JLabel("Area :");
lbl_area.setForeground(c2);
lbl_area.setBounds(40, 260, 150, 30);
lbl_area.setFont(keyFont);
mainPanel.add(lbl_area);

JLabel lbl_area_state = new JLabel(rs.getString("area"));
```

```
lbl_area_state.setBounds(220, 260, 100, 30);
lbl_area_state.setFont(valueFont);
mainPanel.add(lbl_area_state);


JLabel lbl_description = new JLabel("Description :");
lbl_description.setForeground(c2);
lbl_description.setBounds(40, 300, 150, 30);
lbl_description.setFont(keyFont);
mainPanel.add(lbl_description);

JTextArea lbl_desc_state = new JTextArea(rs.getString("description"));
lbl_desc_state.setBounds(220, 300, 200, 60);
lbl_desc_state.setWrapStyleWord(true);
lbl_desc_state.setLineWrap(true);
lbl_desc_state.setOpaque(false);
lbl_desc_state.setEditable(false);
lbl_desc_state.setFont(valueFont);
mainPanel.add(lbl_desc_state);

JLabel lbl_status = new JLabel("Status :");
lbl_status.setForeground(c2);
lbl_status.setBounds(40, 380, 150, 30);
lbl_status.setFont(keyFont);
mainPanel.add(lbl_status);

String status = rs.getString("status");
JLabel lbl_status_state = new JLabel(status.toUpperCase());
lbl_status_state.setBounds(220, 380, 100, 30);
mainPanel.add(lbl_status_state);

JButton btn_delete = new JButton("Delete");
btn_delete.setBorder(new LineBorder(c2));
btn_delete.setBackground(c4);
btn_delete.setForeground(c3);
btn_delete.setBounds(100, 450, 100, 30);
if(!status.equals("solved")) {
   btn_delete.setEnabled(false);
}
btn_delete.addActionListener(e -> {
   handleDelete(id);
});
mainPanel.add(btn_delete);

JButton btn_back = new JButton("Back");
btn_back.setBorder(new LineBorder(c2));
btn_back.setBackground(c1);
```

```
            btn_back.setForeground(c3);
            btn_back.setBounds(250, 450, 100, 30);
            btn_back.addActionListener(e -> {
                dispose();
            });
            mainPanel.add(btn_back);
            add(mainPanel);
        } catch(SQLException ex) {
            System.out.println("exception occurred while showing complaint" + ex);
        }

        setResizable(false);
        setLayout(null);
        setVisible(true);

    }

    void handleDelete(int id){
        int option = JOptionPane.showConfirmDialog(this, "Are you sure do you want to delete 
this Info?");
        if(option == JOptionPane.YES_NO_OPTION){
            try {
                Connection con = DbConnection.getConnection();
                Statement stmt = con.createStatement();
                String query = "delete from complaints where id="+id;
                int result = stmt.executeUpdate(query);
                JOptionPane.showMessageDialog(this, "Complaint deleted successfully.");
                dispose();
                con.close();
            } catch(SQLException ex) {
                System.out.println("Exception occurred while deleteing complaint.  "+ex);
            }
        }
    }

}
```

## LOGIN PAGE (PUBIC/GOVERNMENT OFFICIALS):

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.regex.Pattern;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPasswordField;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.UIManager;
import javax.swing.border.LineBorder;

public class LoginPage extends JFrame
{
   String userType;
   JLabel lbl_passErr;
   LoginPage(String type)
   {
      this.userType = type;
      setSize(570, 500);
      setTitle("Speak Up");
      setBounds(420, 100, 570, 500);

      Color c1=new Color(50, 105, 168);
      Color c2=new Color(0,0,0);
      Color c3=new Color(255, 255, 255);
      Color c4=new Color(93, 253, 146);

      getContentPane().setBackground(c3);

      Font headerfont = new Font("Itc New Baskerville",Font.BOLD,13);
```

```
JLabel lbl_header = new JLabel(type + " Login");
lbl_header.setForeground(c2);
if(type.equals("Public")){
   lbl_header.setBounds(190, 80, 300, 40);
}
else{
   lbl_header.setBounds(120, 80, 300, 40);
}
Font headerFont = new Font("serif", Font.BOLD, 25);
lbl_header.setFont(headerFont);
add(lbl_header);

Font lblFont = new Font("Serif", Font.PLAIN, 18);

JLabel lbl_phone = new JLabel("Phone Number  :");
lbl_phone.setForeground(c2);
lbl_phone.setBounds(120, 150, 200, 30);
lbl_phone.setFont(headerfont);
add(lbl_phone);

JLabel lbl_phoneErr = new JLabel();
lbl_phoneErr.setFont(new Font("serif", Font.PLAIN, 12));
lbl_phoneErr.setBounds(250, 185, 150, 15);
lbl_phoneErr.setForeground(Color.red);


JTextField txt_phone = new JTextField();
txt_phone.setBorder(new LineBorder(c2));
txt_phone.setBounds(250, 150, 150, 30);
txt_phone.addKeyListener(new KeyListener() {
   @Override
   public void keyTyped(KeyEvent e) {

   }

   @Override
   public void keyPressed(KeyEvent e) {
   }

   @Override
   public void keyReleased(KeyEvent e) {
      String phone = txt_phone.getText();
      lbl_phoneErr.setText("Enter a valid phone number.");
      if(!Pattern.matches("^[6-9]{1}[0-9]{0,9}$", phone)){
         add(lbl_phoneErr);
         validate();
         repaint();
```

```
        }
        else{
            remove(lbl_phoneErr);
            validate();
            repaint();
        }
    }
});
add(txt_phone);

JLabel lbl_password = new JLabel("Password  :");
lbl_password.setForeground(c2);
lbl_password.setBounds(120, 210, 120, 30);
lbl_password.setFont(headerfont);
add(lbl_password);

lbl_passErr = new JLabel();
lbl_passErr.setFont(new Font("serif", Font.PLAIN, 12));
lbl_passErr.setBounds(250, 260, 150, 15);
lbl_passErr.setForeground(Color.red);


JPasswordField txt_password = new JPasswordField();
txt_password.setBorder(new LineBorder(c2));
txt_password.setBounds(250, 210, 150, 30);

txt_password.addKeyListener(new KeyListener() {
    @Override
    public void keyTyped(KeyEvent e) {
    }

    @Override
    public void keyPressed(KeyEvent e) {
    }

    @Override
    public void keyReleased(KeyEvent e) {
        char[] password = txt_password.getPassword();
        if(password.length > 0){
            remove(lbl_passErr);
            validate();
            repaint();
        }
    }
});
add(txt_password);
```

```
JCheckBox check_password = new JCheckBox("Show Password");
check_password.setBounds(250, 245, 150, 15);
check_password.setBackground(c3);
check_password.setForeground(c2);
check_password.setFont(headerfont);
check_password.addActionListener((ActionEvent e) -> {
  if(check_password.isSelected())
    txt_password.setEchoChar('\u0000');
  else
    txt_password.setEchoChar((Character)UIManager.get("PasswordField.echoChar"));
});
add(check_password);

JButton btn_submit = new JButton("Submit");
btn_submit.setBackground(c4);
btn_submit.setForeground(c2);
btn_submit.setBorder(new LineBorder(c2));
btn_submit.setBounds(280, 290, 100, 30);
btn_submit.addActionListener((ActionEvent e) -> {
  boolean flag = true;
  String phone = txt_phone.getText();
  char password[] = txt_password.getPassword();
  if(phone.length() < 1){
    lbl_phoneErr.setText("This field cannot be empty");
    add(lbl_phoneErr);
    validate();
    repaint();
    flag = false;
  } else if (!Pattern.matches("^[6-9]{1}[0-9]{9}$", phone)){
    lbl_phoneErr.setText("Enter a valid phone number");
    add(lbl_phoneErr);
    validate();
    repaint();
    flag = false;
  }
  if(password.length < 1){
    lbl_passErr.setText("This field cannot be empty");
    add(lbl_passErr);
    validate();
    repaint();
    flag = false;
  }
  if(flag) {
    verifyUser(phone, password);
  }
});
add(btn_submit);
```

```
        JButton btn_goBack = new JButton("Back");
        btn_goBack.setBorder(new LineBorder(c2));
        btn_goBack.setBackground(c1);
        btn_goBack.setForeground(c3);
        btn_goBack.setBounds(160, 290, 100, 30);

        btn_goBack.addActionListener((ActionEvent e) -> {
            dispose();
            WelcomePage w1 = new WelcomePage();
        });

        add(btn_goBack);
        if(type.equals("Public")){
            JLabel lbl_newUser = new JLabel("Don't have an Account?");
            lbl_newUser.setBounds(100, 393, 160, 20);
            lbl_newUser.setForeground(c2);
            lbl_newUser.setFont(headerfont);
            add(lbl_newUser);

            JButton btn_newUser = new JButton("Sign Up As " + type);
            btn_newUser.setBorder(new LineBorder(c2));
            btn_newUser.setBackground(c1);
            btn_newUser.setForeground(c3);
            btn_newUser.setBounds(260, 390, 180, 30);
            btn_newUser.addActionListener(e -> {
                if(type.equals("Public")){
                    dispose();
                    RegistrationPublic r1 = new RegistrationPublic();
                }
                else {
                    dispose();
                    RegistrationGovtServant r2 = new RegistrationGovtServant();
                }
            });

            add(btn_newUser);
        }

        setResizable(false);
        setLayout(null);
        setVisible(true);
    }

    void verifyUser(String phone, char[] password){
        try{
            Connection con = DbConnection.getConnection();
        Statement stmt = con.createStatement();
        String query, query2;
        if(userType.equals("Public")){
          query = "select * from public where phone='"+phone+"' ";
        } else {
          query = "select * from govt_servant where phone='"+phone+"'";
        }
        ResultSet user = stmt.executeQuery(query);

        String hashedPassword = HashPassword.toHexString(new String(password));
        if(user.next()){
          String actualPassword = user.getString("password");
          if(actualPassword.equals(hashedPassword)){

            if(userType.equals("Public")){
              PublicModel.login(user);
              UserType.setUserType("Public");
              query2 = "insert into logged_in_user(user_id, user_type)
values('"+user.getInt("id")+"', 'public') ";
              int result = stmt.executeUpdate(query2);
              PublicHomePage p1 = new PublicHomePage();
              dispose();
            }
            else {
              GovtModel.login(user);
              UserType.setUserType("Govt. Servant");
              query2 = "insert into logged_in_user(user_id, user_type)
values('"+user.getInt("id")+"', 'govt_servant') ";
              int result = stmt.executeUpdate(query2);
              dispose();
              GovtHomePage g1 = new GovtHomePage();
            }
          }
          else{
            lbl_passErr.setText("Invalid Password");
            add(lbl_passErr);
            validate();
            repaint();
          }
        }
        else{
          JTextArea lbl_noUser = new JTextArea("Account with given Phone Number does
not exist");
          lbl_noUser.setWrapStyleWord(true);
          lbl_noUser.setLineWrap(true);
          lbl_noUser.setOpaque(false);
          lbl_noUser.setEditable(false);
```

```
                lbl_noUser.setBounds(210, 320, 180, 70);
                lbl_noUser.setFont(new Font("serif", Font.PLAIN, 15));
                lbl_noUser.setForeground(Color.red);
                add(lbl_noUser);
                validate();
                repaint();
            }
            con.close();
        } catch(SQLException ex) {
            System.out.println("exception occurred :" + ex);
        }
    }
}
```

## PUBLIC MODULE →

## REGISTRATION PAGE:

```
package speak_up;

import java.util.*;
import javax.swing.*;
import java.awt.*;
import java.awt.Font;
import java.awt.event.*;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.regex.Pattern;
import javax.swing.UIManager;
import javax.swing.border.LineBorder;

public class RegistrationPublic extends JFrame
{
    JLabel l_title,l_name, l_email,l_gender, l_password, l_confpassword, l_address, l_area,
l_city, l_pincode,l_phone;
    JLabel
l_nameerr,l_emailerror,l_passerr,l_confpasserr,l_adderr,l_areaerr,l_cityerr,l_pinerror,l_phper
ror;
    JTextField tf_name,tf_email,tf_address,tf_pincode,tf_phone;
    JButton btn_register, btn_back;
    JPasswordField p_password, p_confpassword;
    JComboBox cb_gender, cb_city, cb_area;
    JTextArea lbl_knownUser = new JTextArea();
    String name, phone, password, confpassword, email, gender, address, area, city, pincode;
    boolean flag = true;

    Color c1=new Color(50, 105, 168);
    Color c2=new Color(0,0,0);
    Color c3=new Color(255, 255, 255);
    Color c4 = new Color (93, 253, 146);

    RegistrationPublic()
    {
        setTitle("Speak Up");
        setSize(600, 700);
        setBounds(350, 70, 600, 700);
```

```
setTitle("Speakp Up");

Font headerfont = new Font("Itc New Baskerville",Font.BOLD,13);

getContentPane().setBackground(c3);

l_title = new JLabel("PUBLIC REGISTRATION");
l_title.setForeground(c2);
l_title.setFont(new Font("Serif", Font.BOLD, 20));
l_title.setBounds(200, 30, 500, 30);
add(l_title);

String gender_combobox[]={"Male","Female","Others"};

l_name = new JLabel("Name  :");
l_name.setForeground(c2);
l_name.setFont(headerfont);
l_name.setBounds(80, 70, 200, 30);
add(l_name);

tf_name = new JTextField();
tf_name.setBorder(new LineBorder(c2));
tf_name.setBounds(300, 70, 200, 30);
tf_name.addKeyListener(new KeyListener() {
    @Override
    public void keyTyped(KeyEvent e) {
    }

    @Override
    public void keyPressed(KeyEvent e) {
    }

    @Override
    public void keyReleased(KeyEvent e) {
        String name = tf_name.getText();
        if(name.length() < 1){
            l_nameerr.setText("This Field cannot be empty");
            add(l_nameerr);
        } else {
            remove(l_nameerr);
        }
        validate();
        repaint();
    }
});
add(tf_name);

l_nameerr = new JLabel();
l_nameerr.setBounds(300, 100, 150, 15);
l_nameerr.setFont(new Font("serif", Font.PLAIN, 11));
l_nameerr.setForeground(Color.red);

l_email = new JLabel("Email-ID  :");
l_email.setForeground(c2);
l_email.setFont(headerfont);
l_email.setBounds(80, 120, 200, 30);
add(l_email);

tf_email = new JTextField();
tf_email.setBorder(new LineBorder(c2));
tf_email.setBounds(300, 120, 200, 30);
tf_email.addKeyListener(new KeyListener() {
        @Override
        public void keyTyped(KeyEvent e) {

        }

        @Override
        public void keyPressed(KeyEvent e) {
        }
        @Override
        public void keyReleased(KeyEvent e) {
            String email = tf_email.getText();
            if(email.length() < 1){
                l_emailerror.setText("This Field cannot be empty");
                add(l_emailerror);
            }
            else{
                remove(l_emailerror);
            }
            validate();
            repaint();
        }
    });
add(tf_email);

l_emailerror = new JLabel();
l_emailerror.setBounds(300, 150, 150, 15);
l_emailerror.setFont(new Font("serif", Font.PLAIN, 11));
l_emailerror.setForeground(Color.red);

l_gender = new JLabel("Gender  :");
l_gender.setFont(headerfont);
l_gender.setBounds(80,170,200,30);
add(l_gender);

cb_gender=new JComboBox(gender_combobox);
cb_gender.setBorder(new LineBorder(c2));
cb_gender.setBackground(c3);
cb_gender.setBounds(300, 170, 200, 30);
```

```
add(cb_gender);

l_password = new JLabel("Password  :");
l_password.setForeground(c2);
l_password.setFont(headerfont);
l_password.setBounds(80, 220, 200, 30);
add(l_password);

p_password = new JPasswordField();
p_password.setBorder(new LineBorder(c2));
p_password.setBounds(300,220,200,30);
p_password.addFocusListener(new FocusListener() {
   @Override
   public void focusGained(FocusEvent e) {
      if(flag){
         showMessage();
         flag = false;
      }
   }

   @Override
   public void focusLost(FocusEvent e) {
      flag = true;
   }
});
p_password.addKeyListener(new KeyListener() {
   @Override
   public void keyTyped(KeyEvent e) {
   }

   @Override
   public void keyPressed(KeyEvent e) {
   }

   @Override
   public void keyReleased(KeyEvent e) {
      String password = new String(p_password.getPassword());
      if(password.length() < 1){
         l_passerr.setText("This Field cannot be empty");
         add(l_passerr);
      } else {
         remove(l_passerr);
      }
      validate();
      repaint();
   }
});
add(p_password);

l_passerr = new JLabel();
```

```
l_passerr.setBounds(420, 250, 150, 15);
l_passerr.setFont(new Font("serif", Font.PLAIN, 11));
l_passerr.setForeground(Color.red);

JCheckBox c_password = new JCheckBox("Show Password");
c_password.setBackground(c3);
c_password.setForeground(c2);
c_password.setBounds(300,250, 120, 15);
c_password.setFont(new Font("serif", Font.PLAIN, 13));
c_password.addActionListener((ActionEvent e) -> {
   if(c_password.isSelected())
      p_password.setEchoChar('\u0000');
   else
      p_password.setEchoChar((Character)UIManager.get("PasswordField.echoChar"));
});
add(c_password);

l_confpassword = new JLabel("Confirm Password  :");
l_confpassword.setForeground(c2);
l_confpassword.setFont(headerfont);
l_confpassword.setBounds(80, 270, 200, 30);
add(l_confpassword);

p_confpassword = new JPasswordField();
p_confpassword.setBorder(new LineBorder(c2));
p_confpassword.setBounds(300, 270, 200, 30);
p_confpassword .addKeyListener(new KeyListener() {
   @Override
   public void keyTyped(KeyEvent e) {
   }

   @Override
   public void keyPressed(KeyEvent e) {
   }

   @Override
   public void keyReleased(KeyEvent e) {
      String confpassword = new String(p_confpassword .getPassword());
      if(confpassword .length() < 1){
         l_confpasserr .setText("This Field cannot be empty");
         add(l_confpasserr);
      } else {
         remove(l_confpasserr);
      }
      validate();
      repaint();
   }
});
add(p_confpassword);
```

```
        l_confpasserr = new JLabel();
        l_confpasserr.setBounds(420, 300, 150, 15);
        l_confpasserr.setFont(new Font("serif", Font.PLAIN, 11));
        l_confpasserr.setForeground(Color.red);

        JCheckBox c_confpassword = new JCheckBox("Show Password");
        c_confpassword.setBackground(c3);
        c_confpassword.setForeground(c2);
        c_confpassword.setBounds(300, 300, 120, 15);
        c_confpassword.setFont(new Font("serif", Font.PLAIN, 13));
        c_confpassword.addActionListener((ActionEvent e) -> {
            if(c_confpassword.isSelected())
                p_confpassword.setEchoChar('\u0000');
            else

p_confpassword.setEchoChar((Character)UIManager.get("PasswordField.echoChar"));
        });
        add(c_confpassword);

        l_address = new JLabel("Address  :");
        l_address.setForeground(c2);
        l_address.setFont(headerfont);
        l_address.setBounds(80, 320, 200, 30);
        add(l_address);

        tf_address = new JTextField();
        tf_address.setBorder(new LineBorder(c2));
        tf_address.setBounds(300, 320, 200, 30);
        tf_address.addKeyListener(new KeyListener() {
            @Override
            public void keyTyped(KeyEvent e) {
            }

            @Override
            public void keyPressed(KeyEvent e) {
            }

            @Override
            public void keyReleased(KeyEvent e) {
                String address = tf_address.getText();
                if(address.length() < 1){
                    l_adderr.setText("This Field cannot be empty");
                    add(l_adderr);
                } else {
                    remove(l_adderr);
                }
                validate();
                repaint();
            }
        });
```

```
add(tf_address);

l_adderr = new JLabel();
l_adderr.setBounds(300, 350, 150, 15);
l_adderr.setFont(new Font("serif", Font.PLAIN, 11));
l_adderr.setForeground(Color.red);

l_area = new JLabel("Area  :");
l_area.setForeground(c2);
l_area.setFont(headerfont);
l_area.setBounds(80, 370, 200, 30);
add(l_area);

String areas[] = {"Area 1", "Area 2"};
cb_area = new JComboBox(areas);
cb_area.setBorder(new LineBorder(c2));
cb_area.setBackground(c3);
cb_area.setBounds(300, 370, 200, 30);
add(cb_area);

l_city = new JLabel("City  :");
l_city.setForeground(c2);
l_city.setFont(headerfont);
l_city.setBounds(80, 420, 200, 30);
add(l_city);

String cities[] = {"City 1", "City 2"};

cb_city = new JComboBox(cities);
cb_city.setBorder(new LineBorder(c2));
cb_city.setBackground(c3);
cb_city.setBounds(300, 420, 200, 30);
add(cb_city);

l_pincode = new JLabel("Pincode  :");
l_pincode.setForeground(c2);
l_pincode.setFont(headerfont);
l_pincode.setBounds(80,470,200,40);
add(l_pincode);

tf_pincode=new JTextField();
tf_pincode.setBorder(new LineBorder(c2));
l_pinerror=new JLabel();
tf_pincode.setBounds(300, 470, 200, 30);
tf_pincode.addKeyListener(new KeyListener() {
        @Override
        public void keyTyped(KeyEvent e) {

        }
        @Override
        public void keyPressed(KeyEvent e) {
        }

        @Override
        public void keyReleased(KeyEvent e) {
            String pincode = tf_pincode.getText();

            if(pincode.length() < 1){
                l_pinerror.setText("This Field cannot be empty");
                add(l_pinerror);
                validate();
                repaint();
            }
            else if (pincode.length() > 6) {
                l_pinerror.setText("Enter a valid pincode.");
                add(l_pinerror);
                validate();
                repaint();

            }
            else{
                remove(l_pinerror);
                validate();
                repaint();
            }
        }
    });
add(tf_pincode);

l_pinerror = new JLabel();
l_pinerror.setBounds(300, 500, 150, 15);
l_pinerror.setFont(new Font("serif", Font.PLAIN, 11));
l_pinerror.setForeground(Color.red);

l_phone =new JLabel("Phone Number  :");
l_phone.setForeground(c2);
l_phone.setFont(headerfont);
l_phone.setBounds(80,520,200,40);
add(l_phone);

tf_phone=new JTextField();
tf_phone.setBorder(new LineBorder(c2));
tf_phone.setBounds(300, 520, 200, 30);
tf_phone.addKeyListener(new KeyListener() {
        @Override
        public void keyTyped(KeyEvent e) {

        }
    @Override
    public void keyPressed(KeyEvent e) {
    }

    @Override
    public void keyReleased(KeyEvent e) {
      String phone = tf_phone.getText();

      if(phone.length() <1){
        l_phperror.setText("This Field cannot be empty");
        add(l_phperror);
      }
      else if (!Pattern.matches("^[6-9]{1}[0-9]{0,9}$", phone)) {
        l_phperror.setText("Enter a valid phone number");
        add(l_phperror);
      }
      else{
        remove(l_phperror);
      }
      validate();
      repaint();
    }
});
add(tf_phone);

l_phperror = new JLabel();
l_phperror.setBounds(300, 550, 150, 15);
l_phperror.setFont(new Font("serif", Font.PLAIN, 11));
l_phperror.setForeground(Color.red);

btn_register = new JButton("Register");
btn_register.setBackground(c4);
btn_register.setForeground(c2);
btn_register.setBorder(new LineBorder(c2));
btn_register.setFont(headerfont);
btn_register.setBounds(320, 570, 100, 30);
btn_register.addActionListener(e -> {
  handleSubmit();
});
add(btn_register);

btn_back = new JButton("Back");
btn_back.setBorder(new LineBorder(c2));
btn_back.setBackground(c1);
btn_back.setForeground(c3);
btn_back.setFont(headerfont);
btn_back.setBounds(150, 570, 100, 30);
btn_back.addActionListener( e -> {
  dispose();
  WelcomePage w3 = new WelcomePage();
```

```
        });
        add(btn_back);

        setResizable(false);
        setLayout(null);
        setVisible(true);
    }

    void showMessage(){
        UIManager UI=new UIManager();
        UI.put("OptionPane.background", c3);
        UI.put("Panel.background", c3);
        JOptionPane.showMessageDialog(this, "Your Password must contain a lower case and
uppercase alphabet , a number and must be minimum 8 digits long.");
    }


    void handleSubmit()
    {
        name = tf_name.getText();
        email = tf_email.getText();
        gender=cb_gender.getSelectedItem().toString();
        char[] c_password = p_password.getPassword();
        char[] c_confpassword = p_confpassword.getPassword();
        password = new String(c_password);
        confpassword = new String(c_confpassword);
        address = tf_address.getText();
        area = cb_area.getSelectedItem().toString();
        city = cb_city.getSelectedItem().toString();
        pincode = tf_pincode.getText();
        phone = tf_phone.getText();

        boolean flag = CheckValidPublic.checkValid(this, name, email, address, area, city,
pincode, phone, l_nameerr, l_emailerror, l_adderr, l_areaerr, l_cityerr, l_pinerror,
l_phperror);

        if(password.length() < 1){
            l_passerr.setText("This Field cannot be empty.");
            add(l_passerr);
            flag = false;
        } else if (!Pattern.matches("^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z]).{8,32}$", password)){
            p_password.setText("");
            p_confpassword.setText("");
            l_passerr.setText("Password must follow the provided rules");
            add(l_passerr);
            flag = false;
        } else if(!Arrays.equals(c_password, c_confpassword)) {
            l_passerr.setText("The passwords must match.");
            add(l_passerr);
            l_confpasserr.setText("The passwords must match.");
            add(l_confpasserr);
            p_password.setText("");
            p_confpassword.setText("");
            flag = false;
        }
        if(confpassword.length() < 1){
            l_confpasserr.setText("This Field cannot be empty.");
            add(l_confpasserr);
            flag = false;
        }
        if(flag){
            addUser();
        }
        validate();
        repaint();
    }

    void addUser()
    {
        try
        {
            Connection con = DbConnection.getConnection();
            PreparedStatement stmt;
            stmt = con.prepareStatement("select * from public where phone=(?) or email=(?)");
            stmt.setString(1, phone);
            stmt.setString(2, email);
            ResultSet user = stmt.executeQuery();
            if(user.next()){
                remove(lbl_knownUser);
                lbl_knownUser.setWrapStyleWord(true);
                lbl_knownUser.setLineWrap(true);
                lbl_knownUser.setOpaque(false);
                lbl_knownUser.setEditable(false);

                lbl_knownUser.setBounds(170, 600, 250, 70);
                lbl_knownUser.setFont(new Font("serif", Font.PLAIN, 15));
                lbl_knownUser.setForeground(Color.red);

                if(user.getString("phone").equals(phone) &&
user.getString("email").equals(email)) {
                    lbl_knownUser.setText("User with the given phone and email already exists.");
                } else if (user.getString("phone").equals(phone)){
                    lbl_knownUser.setText("User with the given phone already exists.");
                } else if (user.getString("email").equals(email)) {
                    lbl_knownUser.setText("User with the given email already exists.");
                }
                add(lbl_knownUser);
                validate();
                repaint();
            }
            else {
                Statement addStmt = con.createStatement();
                String hashedPassword = HashPassword.toHexString(password);
                String query = "insert into public(name, phone, email, gender, password, address,
pincode, area, city) values('"+name+"', '"+phone+"', '"+email+"', '"+gender+"',
'"+hashedPassword+"', '"+address+"','"+pincode+"', '"+area+"', '"+city+"')";
                int result = addStmt.executeUpdate(query);
                System.out.println("Result :" + result);
                JOptionPane.showMessageDialog(this, "Account sucessfully created! Please Log
In to continue!");
                dispose();
                LoginPage l2 = new LoginPage("Public");
            }
            con.close();

        } catch(HeadlessException | SQLException ex) {
            System.out.println("exception occurred :" + ex);
        }
    }
}
```

## CHECK VALID PUBLIC:

```
package speak_up;

import java.util.regex.Pattern;
import javax.swing.JFrame;
import javax.swing.JLabel;

public class CheckValidPublic {

    static boolean checkValid(JFrame f, String name, String email, String address, String area,
String city, String pincode, String phone,
            JLabel l_nameerr, JLabel l_emailerror, JLabel l_adderr, JLabel l_areaerr, JLabel
l_cityerr, JLabel l_pinerror, JLabel l_phperror  ){

        boolean flag = true;
        if(name.length() < 1){
            l_nameerr.setText("This Field cannot be empty.");
            f.add(l_nameerr);
            flag = false;
        }
        if(email.length() < 1){
            l_emailerror.setText("This Field cannot be empty.");
            f.add(l_emailerror);
            flag = false;
        } else if(!Pattern.matches("^[\\w-\\.]+@([\\w-]+\\.)+[\\w-]{2,4}$", email)){
            l_emailerror.setText("Enter a valid email ID.");
            f.add(l_emailerror);
            flag = false;
        }


        if(address.length() < 1){
            l_adderr.setText("This Field cannot be empty.");
            f.add(l_adderr);
            flag = false;
        }

        if(area.length() < 1){
            l_areaerr.setText("This Field cannot be empty.");
            f.add(l_areaerr);
            flag = false;
        }

        if(city.length() < 1){
            l_cityerr.setText("This Field cannot be empty.");
            f.add(l_cityerr);
            flag = false;
        }

        if(pincode.length() < 1){
            l_pinerror.setText("This Field cannot be empty.");
            f.add(l_pinerror);
            flag = false;
        } else if (!Pattern.matches("\\d{0}\\d{6}", pincode)) {
            l_pinerror.setText("Enter a valid pincode");
            f.add(l_pinerror);
            flag = false;
        }

        if(phone.length() < 1){
            l_phperror.setText("This Field cannot be empty.");
            f.add(l_phperror);
            flag = false;
        } else if (!Pattern.matches("^[6-9]{1}[0-9]{9}$", phone)) {
            l_phperror.setText("Enter a valid phone number");
            f.add(l_phperror);
            flag = false;
        }
        return flag;
    }
}
```

## HOME PAGE:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.border.LineBorder;

public class PublicHomePage extends JFrame{

    public PublicHomePage()
    {
        setBounds(300, 85, WIDTH, HEIGHT);
        setSize(570,500);
        setTitle("Speak Up");

        Color c1=new Color(50, 105, 168);
        Color c2=new Color(0,0,0);
        Color c3=new Color(255, 255, 255);

        getContentPane().setBackground(c3);

        JPanel MainPanel = new JPanel();
        MainPanel.setBounds(400, 10, 150, 150);
        MainPanel.setBackground(c3);
        MainPanel.setLayout(null);

        JLabel lbl_more1 = new JLabel("___");
        JLabel lbl_more2 = new JLabel("___");
        JLabel lbl_more3 = new JLabel("___");
        lbl_more1.setBounds(80, 1, 30, 15);
        lbl_more2.setBounds(80, 6, 30, 15);
        lbl_more3.setBounds(80, 11, 30, 15);

        MainPanel.add(lbl_more1);
        MainPanel.add(lbl_more2);
        MainPanel.add(lbl_more3);

        JPanel panel = new JPanel();
        panel.setBackground(c3);
        panel.setBounds(30, 25, 120, 130);

        JButton btn_profile = new JButton("Profile");
        btn_profile.setBorder(new LineBorder(c2));
btn_profile.setBackground(c1);
btn_profile.setForeground(c3);
btn_profile.setBounds(10, 10, 100, 20);
btn_profile.addActionListener(e -> {
   dispose();
   PublicProfile p1 = new PublicProfile();
});
panel.add(btn_profile);

JButton btn_mycomplaints = new JButton("My Complaints");
btn_mycomplaints.setBorder(new LineBorder(c2));
btn_mycomplaints.setBackground(c1);
btn_mycomplaints.setForeground(c3);
btn_mycomplaints.setBounds(10, 80, 100, 20);
btn_mycomplaints.addActionListener(e -> {
   MyComplaints m1 = new MyComplaints();
});
panel.add(btn_mycomplaints);

JButton btn_logout = new JButton("Logout");
btn_logout.setBorder(new LineBorder(c2));
btn_logout.setBackground(c1);
btn_logout.setForeground(c3);
btn_logout.setBounds(10, 150, 100, 20);
btn_logout.addActionListener(e -> {
   PublicModel.logout();
   dispose();
   WelcomePage w1 = new WelcomePage();
});
panel.add(btn_logout);

lbl_more2.addMouseListener(new MouseListener() {
   @Override
   public void mouseClicked(MouseEvent e) {
      if(panel.isVisible()){
         panel.setVisible(false);
         MainPanel.remove(panel);
         validate();
         repaint();
      }
      else {
         panel.setVisible(true);
         MainPanel.add(panel);
         validate();
         repaint();
      }
   }

   @Override
   public void mousePressed(MouseEvent e) {
   }
```

```
    @Override
    public void mouseReleased(MouseEvent e) {
    }

    @Override
    public void mouseEntered(MouseEvent e) {
        panel.setVisible(true);
        MainPanel.add(panel);
        validate();
        repaint();
    }

    @Override
    public void mouseExited(MouseEvent e) {
    }
});
add(MainPanel);

addMouseListener(new MouseListener() {
    @Override
    public void mouseClicked(MouseEvent e) {
        panel.setVisible(false);
        MainPanel.remove(panel);
        validate();
        repaint();
    }

    @Override
    public void mousePressed(MouseEvent e) {}

    @Override
    public void mouseReleased(MouseEvent e) {}

    @Override
    public void mouseEntered(MouseEvent e) {}

    @Override
    public void mouseExited(MouseEvent e) {}
});

JLabel lbl_header = new JLabel("Hi, "+PublicModel.name+"!");
lbl_header.setForeground(c2);
lbl_header.setBounds(180, 40, 500, 80);
Font headerfont = new Font("serif", Font.BOLD, 25);
lbl_header.setFont(headerfont);
add(lbl_header);

JButton btn_ginfo = new JButton("Show Announcements");
btn_ginfo.setBorder(new LineBorder(c2));
btn_ginfo.setBackground(c1);
```

```
        btn_ginfo.setForeground(c3);
        btn_ginfo.setBounds(180, 180, 170, 40);
        add(btn_ginfo);

        JButton btn_complaint = new JButton("Make Complaint");
        btn_complaint.setBorder(new LineBorder(c2));
        btn_complaint.setBackground(c1);
        btn_complaint.setForeground(c3);
        btn_complaint.setBounds(190, 300, 150, 40);
        add(btn_complaint);

        btn_ginfo.addActionListener(e -> {
            PublicInformation l1 = new PublicInformation();
        });

        btn_complaint.addActionListener(e -> {
            PublicComplaints l2 = new PublicComplaints();
        });

        setLayout(null);
        setResizable(false);
        setVisible(true);
    }
}
```

## PUBLIC MODEL:

```
package speak_up;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class PublicModel {
    static ResultSet user;
    static int id;
    static String name;
    static String email;
    static String phone;
    static String gender;
    static String address;
    static String area;
    static String city;
    static String pincode;

    static void login(ResultSet logged_user){
        try{
            user = logged_user;
            id = logged_user.getInt("id");
            name = logged_user.getString("name");
            email = logged_user.getString("email");
            phone = logged_user.getString("phone");
            gender = logged_user.getString("gender");
            address = logged_user.getString("address");
            area = logged_user.getString("area");
            city = logged_user.getString("city");
            pincode = logged_user.getString("pincode");
        } catch(SQLException ex) {
            System.out.println("Exception occurred while setting user model "+ex);
        }
    }

    static void logout(){
        try {
            Connection con = DbConnection.getConnection();
            Statement stmt = con.createStatement();
            String query = "delete from logged_in_user";
            int result = stmt.executeUpdate(query);
            con.close();
        } catch(SQLException ex){
            System.out.println("Exception occurred while logging out user");
        }
        name = "";
        email = "";
        phone = "";
        gender = "";
        address = "";
        area = "";
        city = "";
        pincode = "";
    }
}
```

## PUBLIC PROFILE:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.UIManager;
import javax.swing.border.LineBorder;

public class PublicProfile extends JFrame{

    JLabel lbl_header, lbl_name, lbl_phone, lbl_email, lbl_gender, lbl_address, lbl_area,
lbl_city, lbl_pincode;
    JTextField txt_name, txt_phone, txt_email, txt_gender,txt_address, txt_pincode;
    JButton btn_edit, btn_save, btn_cancel, btn_back;
    String name, area, email, city, address, pincode, phone;
    JLabel err_name, err_phone, err_email, err_city, err_address, err_area, err_pincode;
    JComboBox cb_area, cb_city;
    JTextArea err_phone2 = new JTextArea();
    JTextArea err_email2 = new JTextArea();

    Color c1=new Color(50, 105, 168);
    Color c2=new Color(0,0,0);
    Color c3=new Color(255, 255, 255);
    Color c4 =new Color(255, 0, 0);
    Color c5 = new Color(19, 173, 173);

    UIManager UI=new UIManager();

    public PublicProfile()
    {
       setTitle("Speak Up");
       setBounds(350, 70, 600, 500);
       setSize(600, 600);

       getContentPane().setBackground(c3);

       lbl_header = new JLabel("Profile");
       lbl_header.setForeground(c2);
       lbl_header.setFont(new Font("serif", Font.BOLD, 35));
       lbl_header.setBounds(250, 30, 150, 50);
       add(lbl_header);

       btn_back = new JButton("Back");
       btn_back.setBorder(new LineBorder(c2));
       btn_back.setBackground(c1);
       btn_back.setForeground(c3);
       btn_back.setBounds(20, 10, 80, 30);
       btn_back.addActionListener(e -> {
          dispose();
          PublicHomePage p1 = new PublicHomePage();
       });

       add(btn_back);


       Font simpleFont= new Font("serif", Font.PLAIN, 20);

       lbl_name = new JLabel("Name  :");
       lbl_name.setForeground(c2);
       lbl_name.setBounds(20, 100, 100, 25);
       lbl_name.setFont(simpleFont);
       add(lbl_name);

       txt_name = new JTextField(PublicModel.name);
       txt_name.setBorder(new LineBorder(c2));
       txt_name.setBounds(110, 100, 150, 30);
       txt_name.setEditable(false);
       txt_name.addKeyListener(new KeyListener() {
          @Override
          public void keyTyped(KeyEvent e) {
          }

          @Override
```

```
        public void keyPressed(KeyEvent e) {
        }

        @Override
        public void keyReleased(KeyEvent e) {
            String name = txt_name.getText();
            if(name.length() < 1){
                err_name.setText("This Field cannot be empty.");
                add(err_name);
            } else {
                remove(err_name);
            }
            validate();
            repaint();
        }
    });

    add(txt_name);

    Font err_font = new Font("serif", Font.PLAIN, 12);

    err_name = new JLabel();
    err_name.setBounds(110, 140, 150, 15);
    err_name.setFont(err_font);
    err_name.setForeground(Color.red);

    lbl_phone = new JLabel("Phone Number :");
    lbl_phone.setForeground(c2);
    lbl_phone.setBounds(300, 100, 150, 25);
    lbl_phone.setFont(simpleFont);
    add(lbl_phone);

    txt_phone = new JTextField(PublicModel.phone);
    txt_phone.setBorder(new LineBorder(c2));
    txt_phone.setBounds(440, 100, 100, 30);
    txt_phone.setEditable(false);
    txt_phone.addKeyListener(new KeyListener() {
        @Override
        public void keyTyped(KeyEvent e) {
        }

        @Override
        public void keyPressed(KeyEvent e) {
        }

        @Override
        public void keyReleased(KeyEvent e) {
            String phone = txt_phone.getText();
            remove(err_phone2);
            if(phone.length() < 1){
```

```
            err_phone.setText("This Field cannot be empty.");
            add(err_phone);
        } else {
            remove(err_phone);
        }
        validate();
        repaint();
    }
});
add(txt_phone);

err_phone = new JLabel();
err_phone.setBounds(420, 135, 150, 15);
err_phone.setFont(err_font);
err_phone.setForeground(Color.red);

lbl_email = new JLabel("Email-ID :");
lbl_email.setForeground(c2);
lbl_email.setBounds(20, 170, 100, 25);
lbl_email.setFont(simpleFont);
add(lbl_email);

txt_email = new JTextField(PublicModel.email);
txt_email.setBorder(new LineBorder(c2));
txt_email.setBounds(110, 170, 150, 30);
txt_email.setEditable(false);
txt_email.addKeyListener(new KeyListener() {
    @Override
    public void keyTyped(KeyEvent e) {
    }

    @Override
    public void keyPressed(KeyEvent e) {
    }

    @Override
    public void keyReleased(KeyEvent e) {
        String email = txt_email.getText();
        remove(err_email2);
        if(email.length() < 1){
            err_email.setText("This Field cannot be empty.");
            add(err_email);
        } else {
            remove(err_email);
        }
        validate();
        repaint();
    }
});
add(txt_email);
```

```
err_email = new JLabel();
err_email.setBounds(110, 205, 150, 15);
err_email.setFont(err_font);
err_email.setForeground(Color.red);

lbl_gender = new JLabel("Gender :");
lbl_gender.setForeground(c2);
lbl_gender.setBounds(300, 170, 100, 25);
lbl_gender.setFont(simpleFont);
add(lbl_gender);

txt_gender = new JTextField(PublicModel.gender);
txt_gender.setBorder(new LineBorder(c2));
txt_gender.setBounds(380, 170, 100, 30);
txt_gender.setEditable(false);
add(txt_gender);

lbl_address = new JLabel("Address :");
lbl_address.setForeground(c2);
lbl_address.setBounds(20, 240, 100, 25);
lbl_address.setFont(simpleFont);
add(lbl_address);

txt_address = new JTextField(PublicModel.address);
txt_address.setBorder(new LineBorder(c2));
txt_address.setBounds(110, 240, 300, 30);
txt_address.setEditable(false);
txt_address.addKeyListener(new KeyListener() {
   @Override
   public void keyTyped(KeyEvent e) {
   }

   @Override
   public void keyPressed(KeyEvent e) {
   }

   @Override
   public void keyReleased(KeyEvent e) {
      String address = txt_address.getText();
      if(address.length() < 1){
         err_address.setText("This Field cannot be empty.");
         add(err_address);
      } else {
         remove(err_address);
      }
      validate();
      repaint();
   }
});
```

```
add(txt_address);

err_address = new JLabel();
err_address.setBounds(110, 275, 150, 15);
err_address.setFont(err_font);
err_address.setForeground(Color.red);

lbl_area = new JLabel("Area :");
lbl_area.setForeground(c2);
lbl_area.setBounds(20, 310, 150, 25);
lbl_area.setFont(simpleFont);
add(lbl_area);

String areas[] = {PublicModel.area, "Area 1", "Area 2"};
cb_area = new JComboBox(areas);
cb_area.setBorder(new LineBorder(c2));
cb_area.setBackground(c3);
cb_area.setForeground(c2);
cb_area.setBounds(110, 310, 150, 30);
cb_area.setEnabled(false);
add(cb_area);

err_area = new JLabel();
err_area.setBounds(110, 345, 150, 15);
err_area.setFont(err_font);
err_area.setForeground(Color.red);


lbl_city = new JLabel("City :");
lbl_city.setForeground(c2);
lbl_city.setBounds(300, 310, 150, 25);
lbl_city.setFont(simpleFont);
add(lbl_city);

String cities[] = {PublicModel.city, "City 1", "City 2"};
cb_city = new JComboBox(cities);
cb_city.setBorder(new LineBorder(c2));
cb_city.setBackground(c3);
cb_city.setForeground(c2);
cb_city.setBounds(350, 310, 150, 30);
cb_city.setEnabled(false);
add(cb_city);


err_city = new JLabel();
err_city.setBounds(420, 345, 150, 15);
err_city.setFont(err_font);
err_city.setForeground(Color.red);
```

```
lbl_pincode = new JLabel("Pincode :");
lbl_pincode.setForeground(c2);
lbl_pincode.setBounds(20, 380, 150, 25);
lbl_pincode.setFont(simpleFont);
add(lbl_pincode);

txt_pincode = new JTextField(PublicModel.pincode);
txt_pincode.setBorder(new LineBorder(c2));
txt_pincode.setBounds(110, 380, 150, 30);
txt_pincode.setEditable(false);
txt_pincode.addKeyListener(new KeyListener() {
    @Override
    public void keyTyped(KeyEvent e) {
    }

    @Override
    public void keyPressed(KeyEvent e) {
    }

    @Override
    public void keyReleased(KeyEvent e) {
        String pincode = txt_pincode.getText();
        if(pincode.length() < 1){
            err_pincode.setText("This Field cannot be empty.");
            add(err_pincode);
        } else {
            remove(err_pincode);
        }
        validate();
        repaint();
    }
});
add(txt_pincode);

err_pincode = new JLabel();
err_pincode.setBounds(110, 415, 150, 15);
err_pincode.setFont(err_font);
err_pincode.setForeground(Color.red);


btn_save = new JButton("Save");
btn_save.setBorder(new LineBorder(c2));
btn_save.setBackground(c5);
btn_save.setForeground(c2);
btn_save.setBounds(185, 450, 100, 30);
btn_save.addActionListener(e -> {
    handleSave();
});

btn_cancel = new JButton("Cancel");
```

```
btn_cancel.setBorder(new LineBorder(c2));
btn_cancel.setBackground(c5);
btn_cancel.setForeground(c2);
btn_cancel.setBounds(310, 450, 100, 30);
btn_cancel.addActionListener(e -> {
   dispose();
   PublicProfile p1 = new PublicProfile();
});

btn_edit = new JButton("Edit");
btn_edit.setBorder(new LineBorder(c2));
btn_edit.setBackground(c5);
btn_edit.setForeground(c2);
btn_edit.setBounds(250, 450, 100, 30);
btn_edit.addActionListener(e -> {
   txt_name.setEditable(true);
   txt_phone.setEditable(true);
   txt_email.setEditable(true);
   txt_address.setEditable(true);
   cb_area.setEnabled(true);
   cb_city.setEnabled(true);
   txt_pincode.setEditable(true);
   remove(btn_edit);
   add(btn_save);
   add(btn_cancel);
   validate();
   repaint();
});
add(btn_edit);


JButton btn_changePassword = new JButton("Change Password");
btn_changePassword.setBorder(new LineBorder(c2));
btn_changePassword.setBackground(c1);
btn_changePassword.setForeground(c3);
btn_changePassword.setBounds(120, 500, 150, 30);
btn_changePassword.addActionListener(e -> {
   ChangePassword c9 = new ChangePassword();
});
add(btn_changePassword);

JButton btn_deleteAccount = new JButton("Delete Account");
btn_deleteAccount.setBorder(new LineBorder(c2));
btn_deleteAccount.setBounds(320, 500, 150, 30);
btn_deleteAccount.setBackground(c4);
btn_deleteAccount.setForeground(c3);
btn_deleteAccount.addActionListener(e -> {
   handleDelete();
});
add(btn_deleteAccount);
```

```
        setLayout(null);
        setResizable(false);
        setVisible(true);
    }
    void handleSave(){
        name = txt_name.getText();
        email = txt_email.getText();
        address = txt_address.getText();
        area = cb_area.getSelectedItem().toString();
        city = cb_city.getSelectedItem().toString();
        pincode = txt_pincode.getText();
        phone = txt_phone.getText();

        boolean flag = CheckValidPublic.checkValid(this, name, email, address, area, city,
pincode, phone, err_name, err_email, err_address, err_area, err_city, err_pincode,
err_phone);
        validate();
        repaint();
        if(flag && checkNotExisting()) {
            updateProfile();
        }
    }

    boolean checkNotExisting() {
        boolean f = true;
        try {
            Connection con = DbConnection.getConnection();
            Statement stmt1 = con.createStatement();
            String userQuery = "select * from public where id="+PublicModel.id;
            ResultSet current = stmt1.executeQuery(userQuery);
            current.next();
            if(!phone.equals(current.getString("phone"))){
                Statement stmt2 = con.createStatement();
                String query1 = "select * from public where phone='"+phone+"'";
                ResultSet result1 = stmt2.executeQuery(query1);
                if(result1.next()) {
                    f = false;
                    err_phone2 = new JTextArea("User with the given phone already exists.");
                    err_phone2.setBounds(420, 135, 150, 30);
                    err_phone2.setWrapStyleWord(true);
                    err_phone2.setLineWrap(true);
                    err_phone2.setOpaque(false);
                    err_phone2.setEditable(false);
                    err_phone2.setForeground(Color.red);
                    add(err_phone2);
                    validate();
                    repaint();
                }
            }
            if(!email.equals(current.getString("email"))){
                Statement stmt3 = con.createStatement();
                String query2 = "select * from public where email='"+email+"'";
                ResultSet result2 = stmt3.executeQuery(query2);
                if(result2.next()) {
                    f = false;
                    err_email2 = new JTextArea("User with the given email already exists.");
                    err_email2.setBounds(110, 205, 150, 30);
                    err_email2.setWrapStyleWord(true);
                    err_email2.setLineWrap(true);
                    err_email2.setOpaque(false);
                    err_email2.setEditable(false);
                    err_email2.setForeground(Color.red);
                    add(err_email2);
                    validate();
                    repaint();
                }
            }
            return f;
        } catch(SQLException ex) {
            System.out.println("Exception occurred  "+ex);
        }
        return false;
    }


    void updateProfile() {
        try {
            Connection con = DbConnection.getConnection();
            Statement stmt = con.createStatement();
            String query = "update public set name='"+name+"', phone='"+phone+"',
email='"+email+"', address='"+address+"', area='"+area+"', city='"+city+"',
pincode='"+pincode+"' where id="+PublicModel.id+" ";
            int result = stmt.executeUpdate(query);
            UI.put("OptionPane.background", c3);
            UI.put("Panel.background", c3);
            JOptionPane.showMessageDialog(this, "Profile successfully updated");
            String query2 = "select * from public where id="+PublicModel.id+" ";
            ResultSet user = stmt.executeQuery(query2);
            user.next();
            PublicModel.login(user);
            disableEdit();
            remove(btn_save);
            remove(btn_cancel);
            add(btn_edit);
            validate();
            repaint();
            con.close();
        } catch(Exception ex) {
            System.out.println("Exception in updating profile " + ex);
        }
    }

    void disableEdit(){
        txt_name.setEditable(false);
        txt_phone.setEditable(false);
        txt_email.setEditable(false);
        txt_address.setEditable(false);
        cb_area.setEnabled(false);
        cb_city.setEnabled(false);
        txt_pincode.setEditable(false);

    }

    void handleDelete(){
        int option = JOptionPane.showConfirmDialog(this, "Are you sure to delete your
Account? (This action cannot be undone)");
        if(option == JOptionPane.YES_OPTION){
            try{
               Connection con = DbConnection.getConnection();
                Statement stmt = con.createStatement();
                String query = "delete from public where id="+PublicModel.id;
                int result = stmt.executeUpdate(query);
                UI.put("OptionPane.background", c3);
                UI.put("Panel.background", c3);
                JOptionPane.showMessageDialog(this, "Account deleted!");
                PublicModel.logout();
                dispose();
                WelcomePage w3 = new WelcomePage();
                con.close();
            }catch(SQLException ex) {
                System.out.println("Exception occurred while deletng account. "+ ex);
            }
        }
    }
}
```

# MAKE COMPLAINT:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.UIManager;
import javax.swing.border.LineBorder;

public class PublicComplaints extends JFrame{

   JLabel  l_complaints,l_sector,l_description,information,l_infoerr,l_descerr, l_area;
   JTextField tf_description, tf_area;
   JTextArea tf_info;
   JComboBox cb_sector;
   JButton btn_submit,btn_back;

   Color c1=new Color(50, 105, 168);
   Color c2=new Color(0,0,0);
   Color c3=new Color(255, 255, 255);
   Color c4 = new Color (93, 253, 146);

   PublicComplaints(){

      setTitle("Speak Up");
      setBounds(300, 85, WIDTH, HEIGHT);
      setSize(570, 570);

      getContentPane().setBackground(c3);

      l_complaints = new JLabel("POST YOUR COMPLAINTS");
      l_complaints.setForeground(c2);
      l_complaints.setBounds(115, 10, 400, 80);
      Font headerfont = new Font("serif", Font.BOLD, 25);
      l_complaints.setFont(headerfont);
      add(l_complaints);
```

```
JLabel lbl_sect = new JLabel("Sectors :");
lbl_sect.setForeground(c2);
lbl_sect.setBounds(70, 80, 300, 30);
add(lbl_sect);

String sectors[]={"Corporation","Water","Electricity"};
cb_sector = new JComboBox(sectors);
cb_sector.setBorder(new LineBorder(c2));
cb_sector.setBackground(c3);
cb_sector.setBounds(170, 80, 300, 30);
add(cb_sector);

l_area = new JLabel("Area :");
l_area.setForeground(c2);
l_area.setBounds(70, 130, 150, 30);
add(l_area);

tf_area = new JTextField(PublicModel.area);
tf_area.setBorder(new LineBorder(c2));
tf_area.setBounds(170, 130, 150, 30);
tf_area.setEditable(false);
add(tf_area);


l_description = new JLabel("Description :");
l_description.setForeground(c2);
l_description.setBounds(70, 180, 150, 30);
add(l_description);
l_descerr = new JLabel();

tf_description = new JTextField();
tf_description.setBorder(new LineBorder(c2));
tf_description.setBounds(170, 180, 300, 30);
tf_description.addKeyListener(new KeyListener() {
    @Override
    public void keyTyped(KeyEvent e) {

    }

    @Override
    public void keyPressed(KeyEvent e) {
    }
    @Override
    public void keyReleased(KeyEvent e) {
      String description = tf_description.getText();
      if(description.length() < 1){
        l_descerr.setText("This Field cannot be empty");
        add(l_descerr);
      }
      else{
        remove(l_descerr);
```

```
            }
            validate();
            repaint();
        }
    });
add(tf_description);

l_descerr.setBounds(170, 210, 150, 15);
l_descerr.setFont(new Font("serif", Font.PLAIN, 11));
l_descerr.setForeground(Color.red);

information = new JLabel("Information :");
information.setForeground(c2);
information.setBounds(70, 230, 100, 30);
add(information);


tf_info=new JTextArea();
tf_info.setBorder(new LineBorder(c2));
tf_info.setBounds(170,230,300,200);
tf_info.setWrapStyleWord(true);
tf_info.setLineWrap(true);
tf_info.addKeyListener(new KeyListener() {
   @Override
    public void keyTyped(KeyEvent e) {
    }

    @Override
    public void keyPressed(KeyEvent e) {
    }
    @Override
    public void keyReleased(KeyEvent e) {
        String information = tf_info.getText();
        if(information.length() < 1){
            l_infoerr.setText("This Field cannot be empty");
            add(l_infoerr);
        }
        else{
            remove(l_infoerr);
        }
        validate();
        repaint();
    }
});
add(tf_info);
l_infoerr = new JLabel();
l_infoerr.setBounds(170, 430, 150, 15);
l_infoerr.setFont(new Font("serif", Font.PLAIN, 11));
l_infoerr.setForeground(Color.red);

btn_back = new JButton("Back");
```

```
        btn_back.setBackground(c1);
        btn_back.setForeground(c3);
        btn_back.setBorder(new LineBorder(c2));
        btn_back.setBounds(150, 470, 100, 30);
        add(btn_back);

        btn_submit = new JButton("Post");
        btn_submit.setBackground(c4);
        btn_submit.setForeground(c2);
        btn_submit.setBorder(new LineBorder(c2));
        btn_submit.setBounds(300, 470, 100, 30);
        btn_submit.addActionListener((ActionEvent e) -> {
            handelsubmit();
        });
        add(btn_submit);

        btn_back.addActionListener((ActionEvent e) -> {
            dispose();
        });

        setResizable(false);
        setLayout(null);
        setVisible(true);
    }

    public void handelsubmit()
    {
        String sector=cb_sector.getSelectedItem().toString();
        String description = tf_description.getText();
        String informations = tf_info.getText();

        boolean flag=true;
        if(description.length() < 1){
            l_descerr.setText("This Field cannot be empty.");
            add(l_descerr);
            flag = false;
        }
        if(informations.length() < 1){
            l_infoerr.setText("This Field cannot be empty.");
            add(l_infoerr);
            flag = false;
        }
        if(flag){
            addComplaint(sector,description,informations);
        }
        validate();
        repaint();
    }

    void addComplaint(String sector ,String description, String information)
    {
        try {
          Connection con = DbConnection.getConnection();
          Statement stmt = con.createStatement();
          String query="insert into complaints(sector,description,city, person_id, information, status,
area) values('"+sector+"','"+description+"','"+PublicModel.city+"', "+PublicModel.id+",
'"+information+"', 'posted', '"+PublicModel.area+"');";
          int result = stmt.executeUpdate(query);
          System.out.println("Result :" + result);
          UIManager UI=new UIManager();
          UI.put("OptionPane.background", c3);
          UI.put("Panel.background", c3);
          JOptionPane.showMessageDialog(this, "Your Complaint has been posted Sucessfully.");
          dispose();
          con.close();
        }
        catch(SQLException ex){
          System.out.println("exception occurred :" + ex);
       }
   }
}
```

## MY COMPLAINT:

```
package speak_up;

import java.awt.Color;
import javax.swing.JFrame;
import javax.swing.JButton;
import java.awt.Font;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.SQLException;
import javax.swing.JPanel;
import javax.swing.border.LineBorder;

public class MyComplaints extends JFrame{

    ResultSet rs;
    MyComplaints()
    {
        setSize(550, 500);
        setBounds(300, 85, 550, 500);
        setTitle("Speak Up");
```

```
Color c1=new Color(50, 105, 168);
Color c2=new Color(0,0,0);
Color c3=new Color(255, 255, 255);

getContentPane().setBackground(c3);

try {
   Connection con = DbConnection.getConnection();
   String query1 = "select * from complaints where person_id=(?) ";
   PreparedStatement state = con.prepareStatement(query1);
   state.setInt(1, PublicModel.id);
   System.out.println(PublicModel.id);
   rs = state.executeQuery();
   con.close();
} catch(SQLException exp){
   JOptionPane.showMessageDialog(this,"Error Occured while connecting to database...");
}

JLabel main_lbl = new JLabel("MY COMPLAINTS");
main_lbl.setForeground(c2);
main_lbl.setFont(new Font("Verdana", Font.BOLD, 18));
main_lbl.setBounds(180, 10, 200, 50);
add(main_lbl);
setLayout(new FlowLayout());
try{

   if(rs.next())
   {
      JPanel panel_mainPanel = new JPanel();
      panel_mainPanel.setBackground(c3);
      GridLayout grid = new GridLayout(10, 1);
      panel_mainPanel.setLayout(grid);
      GridLayout g = new GridLayout(1, 4, 20, 20);
      panel_mainPanel.setBounds(30, 80, 500, 300);

      Font headingFont = new Font("serif", Font.BOLD, 17);

      JPanel panel_header = new JPanel(g);
      panel_header.setBackground(c3);
      JLabel lbl_headSector = new JLabel("  Sector");
      lbl_headSector.setForeground(c2);
      lbl_headSector.setFont(headingFont);

      JLabel lbl_headDescription = new JLabel("Description");
      lbl_headDescription.setForeground(c2);
      lbl_headDescription.setFont(headingFont);

      JLabel lbl_headStatus = new JLabel(" Status");
      lbl_headStatus.setForeground(c2);
      lbl_headStatus.setFont(headingFont);

      JLabel lbl_headView = new JLabel("     View");
      lbl_headView.setForeground(c2);
      lbl_headView.setFont(headingFont);
```

```
        panel_header.add(lbl_headSector);
        panel_header.add(lbl_headDescription);
        panel_header.add(lbl_headStatus);
        panel_header.add(lbl_headView);

        panel_mainPanel.add(panel_header);
        do
        {
            JPanel panel = new JPanel(g);
            panel.setBackground(c3);
            panel.setSize(300, 100);
            JLabel lbl_sector = new JLabel(rs.getString("sector").toUpperCase());
            JLabel lbl_status = new JLabel(rs.getString("status").toUpperCase());
            JLabel lbl_description = new JLabel(rs.getString("description"));

            JButton btn_view = new JButton("View");
            btn_view.setBorder(new LineBorder(c2));
            btn_view.setBackground(c1);
            btn_view.setForeground(c3);
            int id = rs.getInt("Id");
            btn_view.addActionListener(e -> {
                    showStatus(id);
            });

            panel.add(lbl_sector);
            panel.add(lbl_description);
            panel.add(lbl_status);
            panel.add(btn_view);
            panel_mainPanel.add(panel);

        } while(rs.next());
        panel_mainPanel.setBackground(c3);
        add(panel_mainPanel);
    }
    else {
        JLabel lbl_empty = new JLabel("You haven't made any complaints.");
        lbl_empty.setForeground(c2);
        lbl_empty.setFont(new Font("serif", Font.PLAIN, 16));
        lbl_empty.setBounds(170, 150, 220, 100);
        add(lbl_empty);
    }
} catch(SQLException ex) {
    System.out.println("Exception while showing complaints. "+ex);
}

JButton btn_back = new JButton("Back");
btn_back.setBorder(new LineBorder(c2));
btn_back.setBackground(c1);
btn_back.setForeground(c3);
btn_back.setBounds(220, 400, 100, 30);
btn_back.addActionListener(e -> {
    dispose();
});
add(btn_back);
```

```
        setResizable(false);
        setLayout(null);
        setVisible(true);

    }

    void showStatus(int id){
        try {
            Connection con = DbConnection.getConnection();
            Statement stmt = con.createStatement();
            String query = "select * from complaints where id="+id;
            ResultSet result = stmt.executeQuery(query);
            result.next();
            dispose();
            ShowMyComplaint s1 = new ShowMyComplaint(result);

        } catch(SQLException ex) {
            System.out.println("Exception in view status "+ ex);
        }
    }
}
```

## VIEW DETAILED COMPLAINT:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.border.LineBorder;

public class ShowMyComplaint extends JFrame{

    public ShowMyComplaint(ResultSet rs)
    {
        setSize(520, 600);
        setBounds(300, 85, 520, 600);
        setTitle("Speak Up");

        Color c1=new Color(50, 105, 168);
Color c2=new Color(0,0,0);
Color c3=new Color(255, 255, 255);
Color c4 =new Color(255, 0, 0);
Color c5 = new Color(19, 173, 173);

getContentPane().setBackground(c3);

try {
   int id = rs.getInt("id");

   Font keyFont = new Font("serif", Font.BOLD, 18);
   Font valueFont = new Font("serif", Font.PLAIN, 18);

   JLabel lbl_head = new JLabel("COMPLAINT");
   lbl_head.setForeground(c2);
   lbl_head.setBounds(170, 20, 150, 30);
   lbl_head.setFont(new Font("serif", Font.BOLD, 23));
   add(lbl_head);

   JPanel mainPanel = new JPanel();
   mainPanel.setBackground(c3);
   mainPanel.setBounds(20, 50, 500, 550);
   mainPanel.setLayout(null);

   JLabel lbl_postedAt = new JLabel("Posted At :");
   lbl_postedAt.setForeground(c2);
   lbl_postedAt.setBounds(40, 20, 150, 30);
   lbl_postedAt.setFont(keyFont);
   mainPanel.add(lbl_postedAt);

   JLabel lbl_postedAt_state = new JLabel(rs.getString("posted_at"));
   lbl_postedAt_state.setBounds(220, 20, 100, 30);
   lbl_postedAt_state.setFont(valueFont);
   mainPanel.add(lbl_postedAt_state);

   JLabel lbl_sector = new JLabel("Sector :");
   lbl_sector.setForeground(c2);
   lbl_sector.setBounds(40, 60, 150, 30);
   lbl_sector.setFont(keyFont);
   mainPanel.add(lbl_sector);

   JLabel lbl_sector_state = new JLabel(rs.getString("sector"));
   lbl_sector_state.setBounds(220, 60, 100, 30);
   lbl_sector_state.setFont(valueFont);
   mainPanel.add(lbl_sector_state);

   JLabel lbl_description = new JLabel("Description :");
   lbl_description.setForeground(c2);
   lbl_description.setBounds(40, 100, 150, 30);
   lbl_description.setFont(keyFont);
   mainPanel.add(lbl_description);
```

```
JTextArea lbl_desc_state = new JTextArea(rs.getString("description"));
lbl_desc_state.setBounds(220, 100, 300, 60);
lbl_desc_state.setWrapStyleWord(true);
lbl_desc_state.setLineWrap(true);
lbl_desc_state.setOpaque(false);
lbl_desc_state.setEditable(false);
lbl_desc_state.setFont(valueFont);
mainPanel.add(lbl_desc_state);

JLabel lbl_info = new JLabel("Information :");
lbl_info.setForeground(c2);
lbl_info.setBounds(40, 140, 150, 30);
lbl_info.setFont(keyFont);
mainPanel.add(lbl_info);

JTextArea lbl_info_state = new JTextArea(rs.getString("information"));
lbl_info_state.setBounds(220, 140, 300, 130);
lbl_info_state.setWrapStyleWord(true);
lbl_info_state.setLineWrap(true);
lbl_info_state.setOpaque(false);
lbl_info_state.setEditable(false);
lbl_info_state.setFont(valueFont);
mainPanel.add(lbl_info_state);

JLabel lbl_status = new JLabel("Status :");
lbl_status.setForeground(c2);
lbl_status.setFont(keyFont);
lbl_status.setBounds(40, 270, 150, 30);
mainPanel.add(lbl_status);

String status = rs.getString("status");
JLabel lbl_status_state = new JLabel();
lbl_status_state.setFont(valueFont);
lbl_status_state.setBounds(220, 270, 240, 30);


if(status.equals("pending")) {
   lbl_status_state.setText("Marked as solved by Official");
} else {
   lbl_status_state.setText(status.toUpperCase());
}
mainPanel.add(lbl_status_state);

JLabel lbl_yourstatus = new JLabel("Conformation Status:");
lbl_yourstatus.setForeground(c2);
lbl_yourstatus.setFont(keyFont);
lbl_yourstatus.setBounds(40, 320, 300, 30);
mainPanel.add(lbl_yourstatus);

JCheckBox check_solved = new JCheckBox("Solved");
```

```
        check_solved.setBackground(c3);
        check_solved.setBounds(240, 375, 80, 20);
        check_solved.setFont(new Font("serif", Font.PLAIN, 17));
        if(status.equals("solved")) {
            check_solved.setSelected(true);
            check_solved.setEnabled(false);
        }

        check_solved.addActionListener(e -> {
            if(check_solved.isSelected()){
                try {
                    Connection con = DbConnection.getConnection();
                    Statement stmt = con.createStatement();
                    String query = "update complaints set status='solved' where id="+id;
                    int result = stmt.executeUpdate(query);
                    lbl_status_state.setText("SOLVED");
                    check_solved.setEnabled(false);
                    validate();
                    repaint();
                } catch(SQLException ex){
                    System.out.println("Exception occurred while setting status "+ex);
                }
            }
        });
        add(check_solved);

        JButton btn_delete = new JButton("Delete");
        btn_delete.setBorder(new LineBorder(c2));
        btn_delete.setBackground(c4);
        btn_delete.setForeground(c3);
        btn_delete.setBounds(100, 430, 100, 30);
        btn_delete.addActionListener(e -> {
            handleDelete(id);
        });
        mainPanel.add(btn_delete);

        JButton btn_back = new JButton("Back");
        btn_back.setBorder(new LineBorder(c2));
        btn_back.setBackground(c1);
        btn_back.setForeground(c3);
        btn_back.setBounds(250, 430, 100, 30);
        btn_back.addActionListener(e -> {
            dispose();
            MyComplaints c6 = new MyComplaints();
        });
        mainPanel.add(btn_back);

        add(mainPanel);
    } catch(SQLException ex) {
        System.out.println("exception occurred while showing my complaint" + ex);
    }
```

```
        setResizable(false);
        setLayout(null);
        setVisible(true);
    }

    void handleDelete(int id) {
        int option = JOptionPane.showConfirmDialog(this, "Are you sure do you want to delete this
complaint?");
        if(option == JOptionPane.YES_NO_OPTION){
            try {
                Connection con = DbConnection.getConnection();
                Statement stmt = con.createStatement();
                String query = "delete from complaints where id="+id;
                int result = stmt.executeUpdate(query);
                JOptionPane.showMessageDialog(this, "Complaint deleted successfully.");
                dispose();
                MyComplaints c1 = new MyComplaints();
            } catch(SQLException ex) {
                System.out.println("Exception occurred while deleteing complaint.  "+ex);
            }
        }
    }
}
```

## SHOW ANNOUNCEMENT:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.awt.GridLayout;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.border.LineBorder;

public class PublicInformation extends JFrame{

    ResultSet e_rs, w_rs, c_rs;

    PublicInformation()
    {
        setBounds(300, 85, WIDTH, HEIGHT);
        setSize(570, 500);
        setTitle("Speak Up");

        Color c1=new Color(50, 105, 168);
        Color c2=new Color(0,0,0);
        Color c3=new Color(255, 255, 255);

        getContentPane().setBackground(c3);

        try
        {
            Connection con = DbConnection.getConnection();
            String query1 = "select * from electricity_board where area='"+PublicModel.area+"'
and city='"+PublicModel.city+"'";
            String query2 = "select * from water_board where area='"+PublicModel.area+"' and
city='"+PublicModel.city+"'";
            String query3 = "select * from corporation_board where area='"+PublicModel.area+"'
and city='"+PublicModel.city+"'";

            Statement stmt1 = con.createStatement();
            Statement stmt2 = con.createStatement();
            Statement stmt3 = con.createStatement();

            e_rs = stmt1.executeQuery(query1);
            w_rs = stmt2.executeQuery(query2);
            c_rs = stmt3.executeQuery(query3);

            con.close();

        } catch(SQLException exp){
            JOptionPane.showMessageDialog(this,"Error Occured while connecting to
database...");
        }

        JLabel main_lbl = new JLabel("ANNOUNCEMENTS");
        main_lbl.setForeground(c2);
        main_lbl.setFont(new Font("Verdana", Font.BOLD, 18));
        main_lbl.setBounds(200, 10, 200, 50);
        add(main_lbl);

        JPanel panel_mainPanel = new JPanel();
        panel_mainPanel.setBackground(c3);
        boolean flag = true;

        panel_mainPanel.setLayout(new GridLayout(10, 1, 25, 2));
        panel_mainPanel.setBounds(25, 50, 500, 300);

        GridLayout grid = new GridLayout(1, 5, 20, 20);
```

```
Font headingFont = new Font("serif", Font.BOLD, 17);

JPanel panel_header = new JPanel(grid);
panel_header.setBackground(c3);

JLabel lbl_headSector = new JLabel("Sector");
lbl_headSector.setForeground(c2);
lbl_headSector.setFont(headingFont);

JLabel lbl_headPostedAt = new JLabel("Posted At");
lbl_headPostedAt.setForeground(c2);
lbl_headPostedAt.setFont(headingFont);


JLabel lbl_headArea = new JLabel("Area");
lbl_headArea.setForeground(c2);
lbl_headArea.setFont(headingFont);

JLabel lbl_headView = new JLabel("View");
lbl_headView.setForeground(c2);
lbl_headArea.setFont(headingFont);

panel_header.add(lbl_headSector);
panel_header.add(lbl_headPostedAt);
panel_header.add(lbl_headArea);
panel_header.add(lbl_headView);

panel_mainPanel.add(panel_header);

try {
  if(e_rs.next()){
    flag = false;
    do {
      JPanel panel = new JPanel(grid);
      panel.setBackground(c3);

      JLabel lbl_sector = new JLabel(e_rs.getString("sector").toUpperCase());
      JLabel lbl_postedAt = new JLabel(e_rs.getString("posted_at"));
      JLabel lbl_area = new JLabel(e_rs.getString("area"));

      JButton btn_view = new JButton("View");
      btn_view.setBorder(new LineBorder(c2));
      btn_view.setBackground(c1);
      btn_view.setForeground(c3);
      int id = e_rs.getInt("Id");
      btn_view.addActionListener(e -> {
          showAnnouncement_E(id);
      });
```

```
          panel.add(lbl_sector);
          panel.add(lbl_postedAt);
          panel.add(lbl_area);
          panel.add(btn_view);
          panel_mainPanel.add(panel);
       } while(e_rs.next());

    }
    if(w_rs.next()){
       flag = false;
       do {
          JPanel panel = new JPanel(grid);
          panel.setBackground(c3);

          JLabel lbl_sector = new JLabel(w_rs.getString("sector").toUpperCase());
          JLabel lbl_postedAt = new JLabel(w_rs.getString("posted_at"));
          JLabel lbl_area = new JLabel(w_rs.getString("area"));

          JButton btn_view = new JButton("View");
          btn_view.setBorder(new LineBorder(c2));
          btn_view.setBackground(c1);
          btn_view.setForeground(c3);
          int id = w_rs.getInt("Id");
          btn_view.addActionListener(e -> {
                showAnnouncement_W(id);
          });

          panel.add(lbl_sector);
          panel.add(lbl_postedAt);
          panel.add(lbl_area);
          panel.add(btn_view);
          panel_mainPanel.add(panel);
       } while(w_rs.next());

    }
    if(c_rs.next()){
       flag = false;
       do {
          JPanel panel = new JPanel(grid);
          panel.setBackground(c3);

          JLabel lbl_sector = new JLabel(c_rs.getString("sector").toUpperCase());
          JLabel lbl_postedAt = new JLabel(c_rs.getString("posted_at"));
          JLabel lbl_area = new JLabel(c_rs.getString("area"));

          JButton btn_view = new JButton("View");
          btn_view.setBorder(new LineBorder(c2));
          btn_view.setBackground(c1);
          btn_view.setForeground(c3);
          int id = c_rs.getInt("Id");
```

```
                btn_view.addActionListener(e -> {
                    showAnnouncement_C(id);
                });

                panel.add(lbl_sector);
                panel.add(lbl_postedAt);
                panel.add(lbl_area);
                panel.add(btn_view);
                panel_mainPanel.add(panel);
            } while(c_rs.next());

        }
        if(flag) {
            JLabel lbl_empty = new JLabel("There are no announcements.");
            lbl_empty.setFont(new Font("serif", Font.PLAIN, 17));
            lbl_empty.setBounds(180, 150, 200, 100);
            add(lbl_empty);
        }
    } catch(SQLException ex) {
        System.out.println("Exception while showing announcements. "+ex);
    }
    JButton btn_back = new JButton("Back");
    btn_back.setBorder(new LineBorder(c2));
    btn_back.setBounds(220, 400, 100, 30);
    btn_back.setBackground(c1);
    btn_back.setForeground(c3);
    btn_back.addActionListener(e -> {
        dispose();
    });
    add(btn_back);
    add(panel_mainPanel);

    setResizable(false);
    setLayout(null);
    setVisible(true);

  }

  void showAnnouncement_E(int id){
    try {
        Connection con = DbConnection.getConnection();
        Statement stmt = con.createStatement();
        String query = "select e.sector, g.name, info, posted_at, e.area, e.city from
electricity_board e join govt_servant g on g.id=e.govt_official_id where e.id="+id;
        ResultSet result = stmt.executeQuery(query);
        result.next();
        dispose();
        ShowAnnouncement s1 = new ShowAnnouncement(result);
        con.close();
    } catch(SQLException ex) {
        System.out.println("Exception in view announcement "+ ex);
    }
  }

  void showAnnouncement_W(int id){
    try {
        Connection con = DbConnection.getConnection();
        Statement stmt = con.createStatement();
        String query = "select e.sector, g.name, info, posted_at, e.area, e.city from
water_board e join govt_servant g on g.id=e.govt_official_id where e.id="+id;
        ResultSet result = stmt.executeQuery(query);
        result.next();
        dispose();
        ShowAnnouncement s1 = new ShowAnnouncement(result);
        con.close();
    } catch(SQLException ex) {
        System.out.println("Exception in view announcement "+ ex);
    }
  }

  void showAnnouncement_C(int id){
    try {
        Connection con = DbConnection.getConnection();
        Statement stmt = con.createStatement();
        String query = "select e.sector, g.name, info, posted_at, e.area, e.city from
corporation_board e join govt_servant g on g.id=e.govt_official_id where e.id="+id;
        ResultSet result = stmt.executeQuery(query);
        result.next();
        dispose();
        ShowAnnouncement s1 = new ShowAnnouncement(result);
        con.close();
    } catch(SQLException ex) {
        System.out.println("Exception in view announcement "+ ex);
    }
  }
}
```

## VIEW DETAILED ANNOUNCEMENT:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.border.LineBorder;

public class ShowAnnouncement extends JFrame {

  ShowAnnouncement(ResultSet rs)
  {
    setSize(520, 500);
    setBounds(300, 85, 520, 500);
    setTitle("Speak Up");

    Color c1=new Color(50, 105, 168);
    Color c2=new Color(0,0,0);
    Color c3=new Color(255, 255, 255);

    getContentPane().setBackground(c3);

    try {

      Font keyFont = new Font("serif", Font.BOLD, 18);
      Font valueFont = new Font("serif", Font.PLAIN, 18);

      JLabel lbl_head = new JLabel(rs.getString("sector").toUpperCase()+" SECTOR");
      lbl_head.setBounds(120, 20, 300, 30);
      lbl_head.setFont(new Font("serif", Font.BOLD, 23));
      add(lbl_head);

      JPanel mainPanel = new JPanel();
      mainPanel.setBackground(c3);
      mainPanel.setBounds(20, 50, 500, 550);
      mainPanel.setLayout(null);

      JLabel lbl_name = new JLabel("Posted By:");
      lbl_name.setFont(keyFont);
      lbl_name.setBounds(40, 20, 150, 30);
      mainPanel.add(lbl_name);

      JLabel lbl_name_state = new JLabel(rs.getString("name"));
```

```
lbl_name_state.setFont(valueFont);
lbl_name_state.setBounds(220, 20, 240, 30);
mainPanel.add(lbl_name_state);

JLabel lbl_area = new JLabel("Area:");
lbl_area.setBounds(40, 60, 150, 30);
lbl_area.setFont(keyFont);
mainPanel.add(lbl_area);

JLabel lbl_area_state = new JLabel(rs.getString("area"));
lbl_area_state.setBounds(220, 60, 100, 30);
lbl_area_state.setFont(valueFont);
mainPanel.add(lbl_area_state);

JLabel lbl_city = new JLabel("City:");
lbl_city.setBounds(40, 100, 150, 30);
lbl_city.setFont(keyFont);
mainPanel.add(lbl_city);

JLabel lbl_city_state = new JLabel(rs.getString("city"));
lbl_city_state.setBounds(220, 100, 100, 30);
lbl_city_state.setFont(valueFont);
mainPanel.add(lbl_city_state);

JLabel lbl_postedAt = new JLabel("Posted At:");
lbl_postedAt.setBounds(40, 140, 150, 30);
lbl_postedAt.setFont(keyFont);
mainPanel.add(lbl_postedAt);

JLabel lbl_postedAt_state = new JLabel(rs.getString("posted_at"));
lbl_postedAt_state.setBounds(220, 140, 100, 30);
lbl_postedAt_state.setFont(valueFont);
mainPanel.add(lbl_postedAt_state);

JLabel lbl_info = new JLabel("Information:");
lbl_info.setBounds(40, 180, 150, 30);
lbl_info.setFont(keyFont);
mainPanel.add(lbl_info);

JTextArea lbl_info_state = new JTextArea(rs.getString("info"));
lbl_info_state.setBounds(220, 180, 300, 130);
lbl_info_state.setWrapStyleWord(true);
lbl_info_state.setLineWrap(true);
lbl_info_state.setOpaque(false);
lbl_info_state.setEditable(false);
lbl_info_state.setFont(valueFont);
mainPanel.add(lbl_info_state);

JButton btn_back = new JButton("Back");
btn_back.setBorder(new LineBorder(c2));
btn_back.setBackground(c1);
```

```
            btn_back.setForeground(c3);
            btn_back.setBounds(175, 340, 100, 30);
            btn_back.addActionListener(e -> {
                PublicInformation p1 = new PublicInformation();
                dispose();
            });
            mainPanel.add(btn_back);
            add(mainPanel);

        } catch(SQLException ex) {
            System.out.println("exception occurred while showing announcement" + ex);
        }
        setResizable(false);
        setLayout(null);
        setVisible(true);
    }
}
```

## GOVERNMENT OFFICIAL MODULE →

## CHECK VALID GOVERNMENT OFFICIAL:

```
package speak_up;

import java.util.regex.Pattern;
import javax.swing.JFrame;
import javax.swing.JLabel;

public class CheckValidGovt {

    static boolean checkValid(JFrame f, String name, String email, String city, String phone,
            JLabel l_nameerr, JLabel l_emailerr, JLabel l_cityerr, JLabel l_phoneerr  ){

        boolean flag = true;

        if(name.length() < 1){
            l_nameerr.setText("This Field cannot be empty.");
            f.add(l_nameerr);
            flag = false;
        }
        if(email.length() < 1){
            l_emailerr.setText("This Field cannot be empty.");
            f.add(l_emailerr);
            flag = false;
        } else if(!Pattern.matches("^[\\w-\\.]+@([\\w-]+\\.)+[\\w-]{2,4}$", email)){
            l_emailerr.setText("Enter a valid email ID.");
            f.add(l_emailerr);
            flag = false;
        }
        if(phone.length() < 1){
            l_phoneerr.setText("This Field cannot be empty.");
            f.add(l_phoneerr);
            flag = false;
        } else if (!Pattern.matches("^[6-9]{1}[0-9]{9}$", phone)) {
            l_phoneerr.setText("Enter a valid phone number");
            f.add(l_phoneerr);
            flag = false;
        }
        if(city.length() < 1){
            l_cityerr.setText("This Field cannot be empty.");
            f.add(l_cityerr);
            flag = false;
        }
        return flag;
    }
}
```

## HOME PAGE:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.border.LineBorder;

public class GovtHomePage extends JFrame{

    public GovtHomePage()
    {
        setBounds(300, 85, WIDTH, HEIGHT);
        setSize(570, 500);
        setTitle("Speak Up");

        Color c1=new Color(50, 105, 168);
        Color c2=new Color(0,0,0);
        Color c3=new Color(255, 255, 255);

        getContentPane().setBackground(c3);
```

```
JPanel MainPanel = new JPanel();
MainPanel.setBounds(400, 10, 150, 150);
MainPanel.setBackground(c3);
MainPanel.setLayout(null);


JLabel lbl_more1 = new JLabel("___");
JLabel lbl_more2 = new JLabel("___");
JLabel lbl_more3 = new JLabel("___");
lbl_more1.setBounds(80, 1, 30, 15);
lbl_more2.setBounds(80, 6, 30, 15);
lbl_more3.setBounds(80, 11, 30, 15);

MainPanel.add(lbl_more1);
MainPanel.add(lbl_more2);
MainPanel.add(lbl_more3);

JPanel panel = new JPanel();
panel.setBackground(c3);
panel.setBounds(40, 25, 100, 110);

JButton btn_profile = new JButton("Profile");
btn_profile.setBorder(new LineBorder(c2));
btn_profile.setBackground(c1);
btn_profile.setForeground(c3);
btn_profile.setBounds(10, 10, 100, 20);
btn_profile.addActionListener(e -> {
   GovtProfile p1 = new GovtProfile();
});
panel.add(btn_profile);

JButton btn_mycomplaints = new JButton("View Info(s)");
btn_mycomplaints.setBorder(new LineBorder(c2));
btn_mycomplaints.setBackground(c1);
btn_mycomplaints.setForeground(c3);
btn_mycomplaints.setBounds(10, 80, 100, 20);
btn_mycomplaints.addActionListener(e -> {
   MyInformations m1 = new MyInformations();
});
panel.add(btn_mycomplaints);

JButton btn_logout = new JButton("Logout");
btn_logout.setBorder(new LineBorder(c2));
btn_logout.setBackground(c1);
btn_logout.setForeground(c3);
btn_logout.setBounds(10, 150, 100, 20);
btn_logout.addActionListener(e -> {
   GovtModel.logout();
   dispose();
   WelcomePage w1 = new WelcomePage();
```

```
});
panel.add(btn_logout);

lbl_more2.addMouseListener(new MouseListener() {
  @Override
  public void mouseClicked(MouseEvent e) {
    if(panel.isVisible()){
      panel.setVisible(false);
      MainPanel.remove(panel);
      validate();
      repaint();
    }
    else {
      panel.setVisible(true);
      MainPanel.add(panel);
      validate();
      repaint();
    }
  }

  @Override
  public void mousePressed(MouseEvent e) {
  }

  @Override
  public void mouseReleased(MouseEvent e) {
  }

  @Override
  public void mouseEntered(MouseEvent e) {
    panel.setVisible(true);
    MainPanel.add(panel);
    validate();
    repaint();
  }

  @Override
  public void mouseExited(MouseEvent e) {
  }
});
add(MainPanel);

addMouseListener(new MouseListener() {
  @Override
  public void mouseClicked(MouseEvent e) {
    panel.setVisible(false);
    MainPanel.remove(panel);
    validate();
    repaint();
  }
```

```
            @Override
            public void mousePressed(MouseEvent e) {}

            @Override
            public void mouseReleased(MouseEvent e) {}

            @Override
            public void mouseEntered(MouseEvent e) {}

            @Override
            public void mouseExited(MouseEvent e) {}
        });

        JLabel lbl_header = new JLabel("Hi, "+GovtModel.name+"!");
        lbl_header.setBounds(200, 40, 500, 80);
        Font headerfont = new Font("serif", Font.BOLD,25);
        lbl_header.setFont(headerfont);
        add(lbl_header);

        JButton btn_pinfo = new JButton("Post Information");
        btn_pinfo.setBorder(new LineBorder(c2));
        btn_pinfo.setBackground(c1);
        btn_pinfo.setForeground(c3);
        btn_pinfo.setBounds(200, 180, 150, 50);
        btn_pinfo.addActionListener(e-> {
            GovtInformation g1 = new GovtInformation();
        });
        add(btn_pinfo);

        JButton btn_vcomplaints = new JButton("View Complaints");
        btn_vcomplaints.setBorder(new LineBorder(c2));
        btn_vcomplaints.setBackground(c1);
        btn_vcomplaints.setForeground(c3);
        btn_vcomplaints.setBounds(200, 300, 150, 50);
        add(btn_vcomplaints);

        btn_vcomplaints.addActionListener(e -> {
            GovtComplaints l2 = new GovtComplaints();
        });

        setResizable(false);
        setLayout(null);
        setVisible(true);

    }
}
```

## GOVERNMENT OFFICIAL MODEL:

```
package speak_up;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class GovtModel {
   static ResultSet user;
   static int id;
   static String name;
   static String email;
   static String phone;
   static String sector;
   static String unique_id;
   static String city;

   static void login(ResultSet logged_user){
      try{
         user = logged_user;
         id = logged_user.getInt("id");
         name = logged_user.getString("name");
         email = logged_user.getString("email");
         phone = logged_user.getString("phone");
         sector = logged_user.getString("sector");
         unique_id = logged_user.getString("unique_id");
         city = logged_user.getString("city");
      } catch(SQLException ex) {
         System.out.println("Exception occurred while setting user model "+ex);
      }
   }

   static void logout(){
      try {
         Connection con = DbConnection.getConnection();
         Statement stmt = con.createStatement();
         String query = "delete from logged_in_user";
         int result = stmt.executeUpdate(query);
         con.close();
      } catch(SQLException ex){
         System.out.println("Exception occurred while logging out user");
      }
      name = "";
      email = "";
      phone = "";
      sector = "";
      unique_id = "";
      city = "";
   }}
```

## PROFILE:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.border.LineBorder;

public class GovtProfile extends JFrame{

   JLabel lbl_header, lbl_name, lbl_phone, lbl_email, lbl_sector, lbl_city, lbl_unique_id;
   JTextField txt_name, txt_phone, txt_email, txt_unique_id, txt_sector;
   JButton btn_edit, btn_save, btn_cancel, btn_back;
   String name, area, email, city, sector, phone;
   JLabel err_name, err_phone, err_email, err_city;
   JComboBox  cb_city;
   JTextArea err_phone2 = new JTextArea();
   JTextArea err_email2 = new JTextArea();

   Color c1=new Color(50, 105, 168);
   Color c2=new Color(0,0,0);
   Color c3=new Color(255, 255, 255);
   Color c4 =new Color(255, 0, 0);
   Color c5 = new Color(19, 173, 173);

   public GovtProfile(){

      setSize(600, 620);
      setBounds(350, 70, 600, 500);
      setTitle("Speak Up");

      getContentPane().setBackground(c3);

      lbl_header = new JLabel("Profile");
      lbl_header.setForeground(c2);
      lbl_header.setFont(new Font("serif", Font.BOLD, 35));
      lbl_header.setBounds(250, 30, 150, 50);
      add(lbl_header);

      btn_back = new JButton("Back");
      btn_back.setBorder(new LineBorder(c2));
      btn_back.setBackground(c1);
      btn_back.setForeground(c3);
      btn_back.setBounds(20, 10, 80, 30);
      btn_back.addActionListener(e -> {
         dispose();
         GovtHomePage g1 = new GovtHomePage();
      });

      add(btn_back);


      Font simpleFont= new Font("serif", Font.PLAIN, 20);

      lbl_name = new JLabel("Name :");
      lbl_name.setBounds(20, 100, 100, 25);
      lbl_name.setFont(simpleFont);
      add(lbl_name);

      txt_name = new JTextField(GovtModel.name);
      lbl_name.setForeground(c2);
      txt_name.setBorder(new LineBorder(c2));
      txt_name.setBounds(110, 100, 150, 30);
      txt_name.setEditable(false);
      txt_name.addKeyListener(new KeyListener() {
         @Override
         public void keyTyped(KeyEvent e) {
         }

         @Override
         public void keyPressed(KeyEvent e) {
         }

         @Override
         public void keyReleased(KeyEvent e) {
            String name = txt_name.getText();
            if(name.length() < 1){
               err_name.setText("This Field cannot be empty.");
               add(err_name);
            } else {
               remove(err_name);
            }
            validate();
            repaint();
         }
      });
```

```
add(txt_name);

Font err_font = new Font("serif", Font.PLAIN, 12);

err_name = new JLabel();
err_name.setBounds(110, 135, 150, 15);
err_name.setFont(err_font);
err_name.setForeground(Color.red);

lbl_phone = new JLabel("Phone Number:");
lbl_phone.setForeground(c2);
lbl_phone.setBounds(290, 100, 200, 25);
lbl_phone.setFont(simpleFont);
add(lbl_phone);

txt_phone = new JTextField(GovtModel.phone);
txt_phone.setBorder(new LineBorder(c2));
txt_phone.setBounds(420, 100, 100, 30);
txt_phone.setEditable(false);
txt_phone.addKeyListener(new KeyListener() {
   @Override
   public void keyTyped(KeyEvent e) {
   }

   @Override
   public void keyPressed(KeyEvent e) {
   }

   @Override
   public void keyReleased(KeyEvent e) {
      remove(err_phone2);
      String phone = txt_phone.getText();
      if(phone.length() < 1){
         err_phone.setText("This Field cannot be empty.");
         add(err_phone);
      } else {
         remove(err_phone);
      }
      validate();
      repaint();
   }
});

add(txt_phone);

err_phone = new JLabel();
err_phone.setBounds(400, 135, 150, 15);
err_phone.setFont(err_font);
err_phone.setForeground(Color.red);
```

```
lbl_email = new JLabel("Email-ID :");
lbl_email.setForeground(c2);
lbl_email.setBounds(20, 170, 100, 25);
lbl_email.setFont(simpleFont);
add(lbl_email);

txt_email = new JTextField(GovtModel.email);
txt_email.setBorder(new LineBorder(c2));
txt_email.setBounds(110, 170, 150, 30);
txt_email.setEditable(false);
txt_email.addKeyListener(new KeyListener() {
   @Override
   public void keyTyped(KeyEvent e) {
   }

   @Override
   public void keyPressed(KeyEvent e) {
   }

   @Override
   public void keyReleased(KeyEvent e) {
      remove(err_email2);
      String email = txt_email.getText();
      if(email.length() < 1){
         err_email.setText("This Field cannot be empty.");
         add(err_email);
      } else {
         remove(err_email);
      }
      validate();
      repaint();
   }
});

add(txt_email);

err_email = new JLabel();
err_email.setBounds(110, 205, 150, 15);
err_email.setFont(err_font);
err_email.setForeground(Color.red);

lbl_sector = new JLabel("Sector :");
lbl_sector.setForeground(c2);
lbl_sector.setBounds(290, 310, 100, 25);
lbl_sector.setFont(simpleFont);
add(lbl_sector);

txt_sector = new JTextField(GovtModel.sector);
txt_sector.setBorder(new LineBorder(c2));
txt_sector.setBounds(400, 310, 150, 30);
txt_sector.setEditable(false);
add(txt_sector);

lbl_city = new JLabel("City :");
lbl_city.setForeground(c2);
lbl_city.setBounds(20, 310, 150, 25);
lbl_city.setFont(simpleFont);
add(lbl_city);

String cities[] = {GovtModel.city, "City 1", "City 2"};
cb_city = new JComboBox(cities);
cb_city.setBorder(new LineBorder(c2));
cb_city.setBackground(c3);
cb_city.setForeground(c2);
cb_city.setBounds(110, 310, 150, 30);
cb_city.setEnabled(false);
add(cb_city);


err_city = new JLabel();
err_city.setBounds(400, 345, 150, 15);
err_city.setFont(err_font);
err_city.setForeground(Color.red);

lbl_unique_id = new JLabel("Aadhar Number :");
lbl_unique_id.setBounds(20, 240, 200, 25);
lbl_unique_id.setFont(simpleFont);
add(lbl_unique_id);

txt_unique_id = new JTextField(GovtModel.unique_id);
txt_unique_id.setBorder(new LineBorder(c2));
txt_unique_id.setBounds(200, 240, 200, 30);
txt_unique_id.setEditable(false);
add(txt_unique_id);


btn_save = new JButton("Save");
btn_save.setBackground(c5);
btn_save.setForeground(c2);
btn_save.setBounds(185, 370, 100, 30);
btn_save.addActionListener(e -> {
   handleSave();
});

btn_cancel = new JButton("Cancel");
btn_cancel.setBorder(new LineBorder(c2));
btn_cancel.setBackground(c5);
btn_cancel.setForeground(c2);
btn_cancel.setBounds(310, 370, 100, 30);
btn_cancel.addActionListener(e -> {
   dispose();
```

```
            GovtProfile p1 = new GovtProfile();
        });

        btn_edit = new JButton("Edit");
        btn_edit.setBorder(new LineBorder(c2));
        btn_edit.setBackground(c5);
        btn_edit.setForeground(c2);
        btn_edit.setBounds(250, 370, 100, 30);
        btn_edit.addActionListener(e -> {
            txt_name.setEditable(true);
            txt_phone.setEditable(true);
            txt_email.setEditable(true);
            cb_city.setEnabled(true);
            remove(btn_edit);
            add(btn_save);
            add(btn_cancel);
            validate();
            repaint();
        });
        add(btn_edit);

        JButton btn_changePassword = new JButton("Change Password");
        btn_changePassword.setBorder(new LineBorder(c2));
        btn_changePassword.setBackground(c1);
        btn_changePassword.setForeground(c3);
        btn_changePassword.setBounds(120, 420, 150, 30);
        btn_changePassword.addActionListener(e -> {
            ChangePassword c7 = new ChangePassword();
        });
        add(btn_changePassword);

        JButton btn_deleteAccount = new JButton("Delete Account");
        btn_deleteAccount.setBorder(new LineBorder(c2));
        btn_deleteAccount.setBounds(320, 420, 150, 30);
        btn_deleteAccount.setBackground(c4);
        btn_deleteAccount.setForeground(c3);
        btn_deleteAccount.addActionListener(e -> {
            handleDelete();
        });
        add(btn_deleteAccount);

        setResizable(false);
        setLayout(null);
        setVisible(true);
    }
    void handleSave(){
        name = txt_name.getText();
        email = txt_email.getText();
        city = cb_city.getSelectedItem().toString();
        phone = txt_phone.getText();
        boolean flag = CheckValidGovt.checkValid(this, name, email, city, phone, err_name, err_email,
err_city, err_phone);

        validate();
        repaint();
        if(flag && checkNotExisting()) {
            updateProfile();
        }
    }

    boolean checkNotExisting() {
        boolean f = true;
        try {
            Connection con = DbConnection.getConnection();
            Statement stmt1 = con.createStatement();
            String userQuery = "select * from govt_servant where id="+GovtModel.id;
            ResultSet current = stmt1.executeQuery(userQuery);
            current.next();
            if(!phone.equals(current.getString("phone"))){
                Statement stmt2 = con.createStatement();
                String query1 = "select * from govt_servant where phone='"+phone+"'";
                ResultSet result1 = stmt2.executeQuery(query1);
                if(result1.next()) {
                    f = false;
                    err_phone2 = new JTextArea("User with the given phone already exists.");
                    err_phone2.setBounds(420, 135, 150, 30);
                    err_phone2.setWrapStyleWord(true);
                    err_phone2.setLineWrap(true);
                    err_phone2.setOpaque(false);
                    err_phone2.setEditable(false);
                    err_phone2.setForeground(Color.red);
                    add(err_phone2);
                    validate();
                    repaint();
                }
            }
            if(!email.equals(current.getString("email"))){
                Statement stmt3 = con.createStatement();
                String query2 = "select * from govt_servant where email='"+email+"'";
                ResultSet result2 = stmt3.executeQuery(query2);
                if(result2.next()) {
                    f = false;
                    err_email2 = new JTextArea("User with the given email already exists.");
                    err_email2.setBounds(110, 205, 150, 30);
                    err_email2.setWrapStyleWord(true);
                    err_email2.setLineWrap(true);
                    err_email2.setOpaque(false);
                    err_email2.setEditable(false);
                    err_email2.setForeground(Color.red);
                    add(err_email2);
                    validate();
```

```
               repaint();
            }
         }
         return f;
      } catch(SQLException ex) {
         System.out.println("Exception occurred  "+ex);
      }
      return false;
   }


   void updateProfile() {
      try {
         Connection con = DbConnection.getConnection();
         Statement stmt = con.createStatement();
         String query = "update govt_servant set name='"+name+"', phone='"+phone+"',
email='"+email+"', city='"+city+"' where id="+GovtModel.id ;
         int result = stmt.executeUpdate(query);
         JOptionPane.showMessageDialog(this, "Profile successfully updated");
         String query2 = "select * from govt_servant where id="+GovtModel.id;
         ResultSet user = stmt.executeQuery(query2);
         user.next();
         GovtModel.login(user);
         disableEdit();
         remove(btn_save);
         remove(btn_cancel);
         add(btn_edit);
         validate();
         repaint();
         con.close();
      } catch(Exception ex) {
         System.out.println("Exception in updating profile " + ex);
      }
   }

   void disableEdit(){
      txt_name.setEditable(false);
      txt_phone.setEditable(false);
      txt_email.setEditable(false);
      cb_city.setEnabled(false);
   }

   void handleDelete(){
      int option = JOptionPane.showConfirmDialog(this, "Are you sure to delete your Account? (This
action cannot be undone)");
      if(option == JOptionPane.YES_OPTION){
         try{
            Connection con = DbConnection.getConnection();
            Statement stmt = con.createStatement();
            String query = "delete from govt_servant where id="+GovtModel.id;
            int result = stmt.executeUpdate(query);
            JOptionPane.showMessageDialog(this, "Account deleted!");
            GovtModel.logout();
            dispose();
            WelcomePage w3 = new WelcomePage();
            con.close();
        }catch(SQLException ex) {
            System.out.println("Exception occurred while deletng account. "+ ex);
        }
      }
    }
}
```

## POST INFORMATION:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.UIManager;
import javax.swing.border.LineBorder;

public class GovtInformation extends JFrame{

    JTextField txt_sector;
    JTextArea tf_info;
    JLabel l_infoerr, lbl_area ;
    JComboBox cb_area;

    Color c1=new Color(50, 105, 168);
    Color c2=new Color(0,0,0);
    Color c3=new Color(255, 255, 255);
    Color c4 = new Color (93, 253, 146);

    UIManager UI=new UIManager();

    public GovtInformation(){

        setTitle("Speak Up");
```

```
setBounds(300, 85, WIDTH, HEIGHT);
setSize(570, 550);

getContentPane().setBackground(c3);

JLabel lbl_ginfotab = new JLabel("POST INFORMATION");
lbl_ginfotab.setBounds(150, 10, 400, 80);
Font headerfont = new Font("serif", Font.BOLD, 25);
lbl_ginfotab.setFont(headerfont);
add(lbl_ginfotab);

JLabel lbl_sector = new JLabel("Sector : ");
lbl_sector.setForeground(c2);
lbl_sector.setBounds(70, 100, 150, 30);
add(lbl_sector);

txt_sector = new JTextField(GovtModel.sector);
txt_sector.setBorder(new LineBorder(c2));
txt_sector.setForeground(c2);
txt_sector.setBounds(170, 100, 300, 30);
txt_sector.setBackground(c3);
txt_sector.setEditable(false);
add(txt_sector);

lbl_area = new JLabel("Area :");
lbl_area.setForeground(c2);
lbl_area.setBounds(70, 150, 150, 30);
add(lbl_area);

String areas[] = {"Area 1", "Area 2"};
cb_area = new JComboBox(areas);
cb_area.setBorder(new LineBorder(c2));
cb_area.setBackground(c3);
cb_area.setBounds(170, 150, 200, 30);
add(cb_area);


JLabel information = new JLabel("Information :");
lbl_area.setForeground(c2);
information.setBounds(70, 200, 100, 30);
add(information);

tf_info=new JTextArea();
tf_info.setBorder(new LineBorder(c2));
tf_info.setBounds(170,190,300,200);
tf_info.setWrapStyleWord(true);
tf_info.setLineWrap(true);

l_infoerr = new JLabel();
tf_info.addKeyListener(new KeyListener() {
  @Override
```

```
            public void keyTyped(KeyEvent e) {
            }

            @Override
            public void keyPressed(KeyEvent e) {
            }
            @Override
            public void keyReleased(KeyEvent e) {
                String information = tf_info.getText();
                if(information.length() < 1){
                    l_infoerr.setText("This Field cannot be empty");
                    add(l_infoerr);
                }
                else{
                    remove(l_infoerr);
                }
                validate();
                repaint();
            }
        });
        add(tf_info);

        l_infoerr.setBounds(170, 400, 150, 15);
        l_infoerr.setFont(new Font("serif", Font.PLAIN, 11));
        l_infoerr.setForeground(Color.red);

        JButton btn_post = new JButton("Post");
        btn_post.setBackground(c4);
        btn_post.setForeground(c2);
        btn_post.setBorder(new LineBorder(c2));
        btn_post.setBounds(320, 430, 100, 30);
        btn_post.addActionListener((ActionEvent e) -> {
            handelsubmit();
        });
        add(btn_post);

        JButton btn_back = new JButton("Back");
        btn_back.setBorder(new LineBorder(c2));
        btn_back.setBackground(c1);
        btn_back.setForeground(c3);
        btn_back.setBounds(120, 430, 100, 30);
        btn_back.addActionListener((ActionEvent e) -> {
            dispose();
        });
        add(btn_back);

        setResizable(false);
        setLayout(null);
        setVisible(true);
    }
    void handelsubmit()
    {
        String informations = tf_info.getText();
        String area = cb_area.getSelectedItem().toString();
        boolean flag=true;

        if(informations.length() < 1){
            l_infoerr.setText("This Field cannot be empty.");
            add(l_infoerr);
            flag = false;
        }
        if(flag){
            addInformation(informations, area);
        }
        validate();
        repaint();
    }

    void addInformation(String information, String area)
    {
         try {
            Connection con = DbConnection.getConnection();
            Statement stmt = con.createStatement();
            String query="insert into "+GovtModel.sector+"_board(sector,info, govt_official_id, area, city)
values('"+GovtModel.sector+"','"+information+"',"+GovtModel.id+", '"+area+"',
'"+GovtModel.city+"')";
            int result = stmt.executeUpdate(query);
            System.out.println("Result :" + result);
            UI.put("OptionPane.background", c3);
            UI.put("Panel.background", c3);
            JOptionPane.showMessageDialog(this, "Your Information has been posted Sucessfully.");
            dispose();
            con.close();
        }
         catch(SQLException ex){
            System.out.println("exception occurred :" + ex);
        }
    }
}
```

## MY INFORMATION:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.awt.GridLayout;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.border.LineBorder;

public class MyInformations extends JFrame{

  ResultSet rs;

  MyInformations()
  {
    setTitle("Speak Up");
    setSize(550, 500);
    setBounds(300, 85, 550, 500);

    Color c1=new Color(50, 105, 168);
    Color c2=new Color(0,0,0);
    Color c3=new Color(255, 255, 255);

    getContentPane().setBackground(c3);

    try {
      Connection con = DbConnection.getConnection();
      String query1 = "select * from "+GovtModel.sector+"_board where govt_official_id=(?) ";
      PreparedStatement state = con.prepareStatement(query1);
      state.setInt(1, GovtModel.id);
      System.out.println(GovtModel.id);
      rs = state.executeQuery();
      con.close();
    } catch(SQLException exp){
      JOptionPane.showMessageDialog(this,"Error Occured while connecting to database...");
    }

    JLabel main_lbl = new JLabel("POSTED INFORMATIONS");
    main_lbl.setBackground(c3);
    main_lbl.setFont(new Font("Verdana", Font.BOLD, 18));
    main_lbl.setBounds(150, 10, 250, 50);
```

```
add(main_lbl);

try{

   if(rs.next()){
      JPanel panel_mainPanel = new JPanel(new GridLayout(10, 1));
      panel_mainPanel.setBackground(c3);
      panel_mainPanel.setBounds(30, 80, 500, 300);

      GridLayout g = new GridLayout(1, 3, 20, 20);

      Font headingFont = new Font("serif", Font.BOLD, 17);

      JPanel panel_header = new JPanel(g);
      panel_header.setBackground(c3);

      JLabel lbl_headPosted = new JLabel("Posted At");
      lbl_headPosted.setForeground(c2);
      lbl_headPosted.setFont(headingFont);

      JLabel lbl_headArea = new JLabel("Area");
      lbl_headArea.setForeground(c2);
      lbl_headArea.setFont(headingFont);

      JLabel lbl_headView = new JLabel("View");
      lbl_headView.setForeground(c2);

      panel_header.add(lbl_headPosted);
      panel_header.add(lbl_headArea);
      panel_header.add(lbl_headView);

      panel_mainPanel.add(panel_header);
      do {
         JPanel panel = new JPanel(g);
         panel.setBackground(c3);
         panel.setSize(300, 100);

         JLabel lbl_posted = new JLabel(rs.getString("posted_at"));
         JLabel lbl_area = new JLabel(rs.getString("area"));

         JButton btn_view = new JButton("View");
         btn_view.setBorder(new LineBorder(c2));
         btn_view.setBackground(c1);
         btn_view.setForeground(c3);
         int id = rs.getInt("Id");
         btn_view.addActionListener(e -> {
               showInfo(id);
         });

         panel.add(lbl_posted);
         panel.add(lbl_area);
```

```
                panel.add(btn_view);
                panel_mainPanel.add(panel);
            } while(rs.next());

            add(panel_mainPanel);
        }
        else {
            JLabel lbl_empty = new JLabel("You haven't posted any information");
            lbl_empty.setBounds(160, 150, 250, 100);
            lbl_empty.setFont(new Font("serif", Font.PLAIN, 17));
            add(lbl_empty);
        }
    } catch(SQLException ex) {
        System.out.println("Exception while showing complaints. "+ex);
    }
    JButton btn_back = new JButton("Back");
    btn_back.setBorder(new LineBorder(c2));
    btn_back.setBackground(c1);
    btn_back.setForeground(c3);
    btn_back.setBounds(220, 400, 100, 30);
    btn_back.addActionListener(e -> {
        dispose();
    });
    add(btn_back);

    setResizable(false);
    setLayout(null);
    setVisible(true);

  }

  void showInfo(int id){
    try {
        Connection con = DbConnection.getConnection();
        Statement stmt = con.createStatement();
        String query = "select * from "+GovtModel.sector+"_board where id="+id;
        ResultSet result = stmt.executeQuery(query);
        result.next();
        dispose();
        ShowPostedInfo s1 = new ShowPostedInfo(result);
        con.close();
    } catch(SQLException ex) {
        System.out.println("Exception in view status "+ ex);
    }
  }
}
```

## VIEW DETAILED INFORMATION:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.UIManager;
import javax.swing.border.LineBorder;

public class ShowPostedInfo extends JFrame{


   Color c1=new Color(50, 105, 168);
   Color c2=new Color(0,0,0);
   Color c3=new Color(255, 255, 255);
   Color c4 =new Color(255, 0, 0);

   UIManager UI=new UIManager();

   ShowPostedInfo(ResultSet rs)
   {
      setTitle("Speak Up");
      setSize(520, 500);
      setBounds(300, 85, 520, 500);

      getContentPane().setBackground(c3);

      try {
         int id = rs.getInt("id");

         Font keyFont = new Font("serif", Font.BOLD, 18);
         Font valueFont = new Font("serif", Font.PLAIN, 18);

         JLabel lbl_head = new JLabel("INFORMATION");
         lbl_head.setForeground(c2);
         lbl_head.setBounds(180, 20, 200, 30);
         lbl_head.setFont(new Font("serif", Font.BOLD, 23));
         add(lbl_head);

         JPanel mainPanel = new JPanel();
         mainPanel.setBackground(c3);
mainPanel.setBounds(20, 50, 500, 550);
mainPanel.setLayout(null);

JLabel lbl_sector = new JLabel("Sector:");
lbl_sector.setFont(keyFont);
lbl_sector.setBounds(40, 20, 150, 30);
mainPanel.add(lbl_sector);

JLabel lbl_sector_state = new JLabel(rs.getString("sector"));
lbl_sector_state.setFont(valueFont);
lbl_sector_state.setBounds(220, 20, 250, 30);
mainPanel.add(lbl_sector_state);

JLabel lbl_area = new JLabel("Area:");
lbl_area.setBounds(40, 100, 150, 30);
lbl_area.setFont(keyFont);
mainPanel.add(lbl_area);

JLabel lbl_area_state = new JLabel(rs.getString("area"));
lbl_area_state.setBounds(220, 100, 100, 30);
lbl_area_state.setFont(valueFont);
mainPanel.add(lbl_area_state);

JLabel lbl_city = new JLabel("City:");
lbl_city.setBounds(40, 60, 150, 30);
lbl_city.setFont(keyFont);
mainPanel.add(lbl_city);

JLabel lbl_city_state = new JLabel(rs.getString("city"));
lbl_city_state.setBounds(220, 60, 100, 30);
lbl_city_state.setFont(valueFont);
mainPanel.add(lbl_city_state);

JLabel lbl_info = new JLabel("Information:");
lbl_info.setBounds(40, 140, 150, 30);
lbl_info.setFont(keyFont);
mainPanel.add(lbl_info);

JTextArea lbl_info_state = new JTextArea(rs.getString("info"));
lbl_info_state.setBounds(220, 140, 300, 60);
lbl_info_state.setWrapStyleWord(true);
lbl_info_state.setLineWrap(true);
lbl_info_state.setOpaque(false);
lbl_info_state.setEditable(false);
lbl_info_state.setFont(valueFont);
mainPanel.add(lbl_info_state);

JButton btn_delete = new JButton("Delete");
btn_delete.setBorder(new LineBorder(c2));
btn_delete.setBounds(100, 350, 100, 30);
btn_delete.addActionListener(e -> {
```

```
                handleDelete(id);
            });
            btn_delete.setBackground(c4);
            btn_delete.setForeground(c3);
            mainPanel.add(btn_delete);

            JButton btn_back = new JButton("Back");
            btn_back.setBorder(new LineBorder(c2));
            btn_back.setBackground(c1);
            btn_back.setForeground(c3);
            btn_back.setBounds(250, 350, 100, 30);
            btn_back.addActionListener(e -> {
                dispose();
                MyInformations m1 = new MyInformations();
            });
            mainPanel.add(btn_back);

            add(mainPanel);
        } catch(SQLException ex) {
            System.out.println("exception occurred while showing my complaint" + ex);
        }

        setResizable(false);
        setLayout(null);
        setVisible(true);
    }

    void handleDelete(int id) {
        UI.put("OptionPane.background", c3);
        UI.put("Panel.background", c3);
        int option = JOptionPane.showConfirmDialog(this, "Are you sure do you want to delete this
Information?");
        if(option == JOptionPane.YES_NO_OPTION){
            try {
                Connection con = DbConnection.getConnection();
                Statement stmt = con.createStatement();
                String query = "delete from "+GovtModel.sector+"_board where id="+id;
                int result = stmt.executeUpdate(query);
                UI.put("OptionPane.background", c3);
                UI.put("Panel.background", c3);
                JOptionPane.showMessageDialog(this, "Complaint deleted successfully.");
                dispose();
                MyInformations m1 = new MyInformations();
                con.close();
            } catch(SQLException ex) {
                System.out.println("Exception occurred while deleteing complaint.  "+ex);
            }
        }
    }
}
```

## VIEW COMPLAINTS:

```
package speak_up;

import java.awt.Color;
import javax.swing.JFrame;
import javax.swing.JButton;
import java.awt.Font;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.SQLException;
import javax.swing.JPanel;
import javax.swing.border.LineBorder;


public class GovtComplaints extends JFrame{

  ResultSet rs;
  GovtComplaints()
  {
    setSize(550, 500);
    setTitle("Speak Up");
    setBounds(420, 100, 550, 500);

    Color c1=new Color(50, 105, 168);
    Color c2=new Color(0,0,0);
    Color c3=new Color(255, 255, 255);

    getContentPane().setBackground(c3);

    try {
      Connection con = DbConnection.getConnection();
      String query1 = "select c.id, name, sector, c.city, c.area, posted_at from complaints c JOIN
public p on c.person_id=p.id where c.sector=(?) and c.city=(?);";
      PreparedStatement state = con.prepareStatement(query1);
      state.setString(1, GovtModel.sector);
      state.setString(2, GovtModel.city);
      rs = state.executeQuery();
      con.close();
    } catch(SQLException exp){
      JOptionPane.showMessageDialog(this,"Error Occured while connecting to database...");
    }

    JLabel main_lbl = new JLabel("PUBLIC COMPLAINTS");
    main_lbl.setForeground(c2);
    main_lbl.setFont(new Font("Verdana", Font.BOLD, 18));
    main_lbl.setBounds(160, 10, 300, 50);
    add(main_lbl);
    setLayout(new FlowLayout());
    try{

      if(rs.next()){
          JPanel panel_mainPanel = new JPanel();
          panel_mainPanel.setBackground(c3);
          panel_mainPanel.setLayout(new GridLayout(10, 1, 25, 2));
          panel_mainPanel.setBounds(25, 50, 500, 300);

          GridLayout grid = new GridLayout(1, 5, 20, 20);

          Font headingFont = new Font("serif", Font.BOLD, 17);

          JPanel panel_header = new JPanel(grid);
          panel_header.setBackground(c3);

          JLabel lbl_headName = new JLabel("Name");
          lbl_headName.setForeground(c2);
          lbl_headName.setFont(headingFont);

          JLabel lbl_headPostedAt = new JLabel("Posted At");
          lbl_headPostedAt.setForeground(c2);
          lbl_headPostedAt.setFont(headingFont);

          JLabel lbl_headArea = new JLabel("Area");
          lbl_headArea.setForeground(c2);
          lbl_headArea.setFont(headingFont);

          JLabel lbl_headView = new JLabel("View");
          lbl_headArea.setForeground(c2);
          lbl_headArea.setFont(headingFont);

          panel_header.add(lbl_headName);
          panel_header.add(lbl_headPostedAt);
          panel_header.add(lbl_headArea);
          panel_header.add(lbl_headView);

          panel_mainPanel.add(panel_header);
          do {
              JPanel panel = new JPanel(grid);
              panel.setBackground(c3);

              JLabel lbl_name = new JLabel(rs.getString("name"));
              JLabel lbl_sector = new JLabel(rs.getString("posted_at"));
              JLabel lbl_area = new JLabel(rs.getString("Area"));

              JButton btn_view = new JButton("View");
              btn_view.setBorder(new LineBorder(c2));
```

```
                btn_view.setBackground(c1);
                btn_view.setForeground(c3);
                int id = rs.getInt("Id");
                btn_view.addActionListener(e -> {
                   dispose();
                   showComplaint(id);
                });

                panel.add(lbl_name);
                panel.add(lbl_sector);
                panel.add(lbl_area);
                panel.add(btn_view);
                panel_mainPanel.add(panel);
             } while(rs.next());
              add(panel_mainPanel);

          }
          else {
             JLabel lbl_empty = new JLabel("There are no complaints.");
             lbl_empty.setFont(new Font("serif", Font.PLAIN, 17));
             lbl_empty.setBounds(180, 150, 200, 100);
             add(lbl_empty);
          }
       } catch(SQLException ex) {
          System.out.println("Exception while showing complaints. "+ex);
       }

       JButton btn_back = new JButton("Back");
       btn_back.setBorder(new LineBorder(c2));
       btn_back.setBackground(c1);
       btn_back.setForeground(c3);
       btn_back.setBounds(220, 400, 100, 30);
       btn_back.addActionListener(e -> {
          dispose();
       });
       add(btn_back);

       setResizable(false);
       setLayout(null);
       setVisible(true);

    }

    void showComplaint(int id){
       try {
          Connection con = DbConnection.getConnection();
          Statement stmt = con.createStatement();
          String query = "select c.id, name, description, status, phone, information, c.area, sector, c.city
from complaints c JOIN public p on c.person_id=p.id where c.id="+id;
          ResultSet result = stmt.executeQuery(query);
          result.next();
```

```
            ShowComplaint s1 = new ShowComplaint(result);
            con.close();
        } catch(SQLException ex) {
            System.out.println("Exception in view status "+ ex);
        }
    }
}
```

## VIEW DETAILED COMPLAINTS:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.awt.GridLayout;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.border.LineBorder;

public class ShowComplaint extends JFrame{

    public ShowComplaint(ResultSet rs)
    {
        setTitle("Speak Up");
        setSize(500, 600);
        setBounds(400, 100, 500, 600);

        Color c1=new Color(50, 105, 168);
        Color c2=new Color(0,0,0);
        Color c3=new Color(255, 255, 255);
        Color c4 =new Color(255, 0, 0);

        getContentPane().setBackground(c3);

        try {
            int id = rs.getInt("id");

            Font keyFont = new Font("serif", Font.BOLD, 18);
            Font valueFont = new Font("serif", Font.PLAIN, 18);

            JLabel lbl_head = new JLabel("COMPLAINT");
            lbl_head.setForeground(c2);
            lbl_head.setBounds(170, 20, 200, 30);
            lbl_head.setFont(new Font("serif", Font.BOLD, 23));
            add(lbl_head);

            JPanel mainPanel = new JPanel(new GridLayout(8, 2));
            mainPanel.setBackground(c3);
            mainPanel.setBounds(20, 50, 440, 550);
            mainPanel.setLayout(null);

            JLabel lbl_name = new JLabel("Name :");
            lbl_name.setForeground(c2);
            lbl_name.setFont(keyFont);
            lbl_name.setBounds(40, 20, 150, 30);
            mainPanel.add(lbl_name);

            JLabel lbl_name_state = new JLabel(rs.getString("name"));
            lbl_name_state.setBounds(220, 20, 100, 30);
            lbl_name_state.setFont(valueFont);
            mainPanel.add(lbl_name_state);

            JLabel lbl_phone = new JLabel("Phone Number :");
            lbl_phone.setForeground(c2);
            lbl_phone.setBounds(40, 60, 200, 30);
            lbl_phone.setFont(keyFont);
            mainPanel.add(lbl_phone);

            JLabel lbl_phone_state = new JLabel(rs.getString("phone"));
            lbl_phone_state.setBounds(220, 60, 100, 30);
            lbl_phone_state.setFont(valueFont);
            mainPanel.add(lbl_phone_state);

            JLabel lbl_city = new JLabel("City :");
            lbl_city.setForeground(c2);
            lbl_city.setBounds(40, 100, 150, 30);
            lbl_city.setFont(keyFont);
            mainPanel.add(lbl_city);

            JLabel lbl_city_state = new JLabel(rs.getString("city"));
            lbl_city_state.setBounds(220, 100, 100, 30);
            lbl_city_state.setFont(valueFont);
            mainPanel.add(lbl_city_state);

            JLabel lbl_area = new JLabel("Area :");
            lbl_area.setForeground(c2);
            lbl_area.setBounds(40, 140, 150, 30);
            lbl_area.setFont(keyFont);
            mainPanel.add(lbl_area);

            JLabel lbl_area_state = new JLabel(rs.getString("area"));
            lbl_area_state.setBounds(220, 140, 100, 30);
            lbl_area_state.setFont(valueFont);
```

```
mainPanel.add(lbl_area_state);

JLabel lbl_description = new JLabel("Description :");
lbl_description.setForeground(c2);
lbl_description.setBounds(40, 180, 150, 30);
lbl_description.setFont(keyFont);
mainPanel.add(lbl_description);

JTextArea lbl_desc_state = new JTextArea(rs.getString("description"));
lbl_desc_state.setBounds(220, 180, 200, 60);
lbl_desc_state.setWrapStyleWord(true);
lbl_desc_state.setLineWrap(true);
lbl_desc_state.setOpaque(false);
lbl_desc_state.setEditable(false);
lbl_desc_state.setFont(valueFont);
mainPanel.add(lbl_desc_state);

JLabel lbl_info = new JLabel("Information :");
lbl_info.setForeground(c2);
lbl_info.setBounds(40, 220, 150, 30);
lbl_info.setFont(keyFont);
mainPanel.add(lbl_info);

JTextArea lbl_info_state = new JTextArea(rs.getString("information"));
lbl_info_state.setBounds(220, 220, 200, 150);
lbl_info_state.setWrapStyleWord(true);
lbl_info_state.setLineWrap(true);
lbl_info_state.setOpaque(false);
lbl_info_state.setEditable(false);
lbl_info_state.setFont(valueFont);
mainPanel.add(lbl_info_state);

JLabel lbl_status = new JLabel("Status :");
lbl_status.setForeground(c2);
lbl_status.setBounds(40, 320, 150, 30);
lbl_status.setFont(keyFont);
mainPanel.add(lbl_status);

String status = rs.getString("status");

JLabel lbl_status_state = new JLabel(status.toUpperCase());
lbl_status_state.setBounds(220, 320, 100, 30);
mainPanel.add(lbl_status_state);

JLabel lbl_cstatus = new JLabel("Conformation Status:");
lbl_cstatus.setForeground(c2);
lbl_cstatus.setBounds(40, 360, 300, 30);
lbl_cstatus.setFont(keyFont);
mainPanel.add(lbl_cstatus);

JCheckBox check_ack = new JCheckBox("Acknowledge");
```

```
check_ack.setBackground(c3);
check_ack.setFont(valueFont);
check_ack.setBounds(220, 360, 150, 20);
if(status.equals("acknowledged") || status.equals("solved") || status.equals("pending") ){
   check_ack.setSelected(true);
   check_ack.setEnabled(false);
}
check_ack.addActionListener(e -> {
   if(check_ack.isSelected()) {
      try {
         Connection con = DbConnection.getConnection();
         Statement stmt = con.createStatement();
         String query = "update complaints set status='acknowledged' where id=" + id;
         int result = stmt.executeUpdate(query);
         check_ack.setEnabled(false);
         lbl_status_state.setText("Acknowledged");
         validate();
         repaint();
         con.close();
      } catch(SQLException ex) {
         System.out.println("Exception while changing status " + ex);
      }
   }
});
mainPanel.add(check_ack);

JCheckBox check_solved = new JCheckBox("Solved");
check_solved.setBackground(c3);
check_solved.setFont(valueFont);
check_solved.setBounds(220, 380, 100, 20);
if(status.equals("solved") || status.equals("pending")) {
   check_solved.setSelected(true);
   check_solved.setEnabled(false);
}
check_solved.addActionListener(e -> {
   if(check_solved.isSelected()){
      try {
         Connection con = DbConnection.getConnection();
         Statement stmt = con.createStatement();
         String query = "update complaints set status='pending' where id=" + rs.getInt("id");
         int result = stmt.executeUpdate(query);
         check_solved.setEnabled(false);
         lbl_status_state.setText("Pending");
         validate();
         repaint();
         con.close();
      } catch(SQLException ex) {
         System.out.println("Exception while changing status " + ex);
      }
   }
});
```

```
            mainPanel.add(check_solved);

            JButton btn_delete = new JButton("Delete");
            btn_delete.setBorder(new LineBorder(c2));
            btn_delete.setBackground(c4);
            btn_delete.setForeground(c3);
            btn_delete.setBounds(100, 450, 100, 30);
            if(!status.equals("solved")) {
               btn_delete.setEnabled(false);
            }
            btn_delete.addActionListener(e -> {
               handleDelete(id);
            });
            mainPanel.add(btn_delete);

            JButton btn_back = new JButton("Back");
            btn_back.setBorder(new LineBorder(c2));
            btn_back.setBackground(c1);
            btn_back.setForeground(c3);
            btn_back.setBounds(250, 450, 100, 30);
            btn_back.addActionListener(e -> {
               dispose();
               GovtComplaints g3 =new GovtComplaints();
            });
            mainPanel.add(btn_back);

            add(mainPanel);
         } catch(SQLException ex) {
            System.out.println("exception occurred while showing complaint" + ex);
         }

         setResizable(false);
         setLayout(null);
         setVisible(true);
      }
      void handleDelete(int id)
      {
         int option = JOptionPane.showConfirmDialog(this, "Are you sure do you want to delete this
Info?");
         if(option == JOptionPane.YES_NO_OPTION){
            try {
               Connection con = DbConnection.getConnection();
               Statement stmt = con.createStatement();
               String query = "delete from complaints where id="+id;
               int result = stmt.executeUpdate(query);
               JOptionPane.showMessageDialog(this, "Complaint deleted successfully.");
               dispose();
               //GovtHomePage m1 = new GovtHomePage();
               con.close();
            } catch(SQLException ex) {
               System.out.println("Exception occurred while deleteing complaint.  "+ex);
            }
        }
    }
}
```

## CHANGE PASSWORD:

```
package speak_up;

import java.awt.Color;
import java.awt.Font;
import java.awt.event.FocusEvent;
import java.awt.event.FocusListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.regex.Pattern;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPasswordField;
import javax.swing.UIManager;
import javax.swing.border.LineBorder;

public class ChangePassword extends JFrame{

    JLabel lbl_currentPass, lbl_newPass, lbl_confPass;
    JPasswordField txt_currentPass, txt_newPass, txt_confPass;
    JLabel err_current, err_new, err_conf;
    String currentPass, newPass, confPass;

    boolean flag;

    UIManager UI=new UIManager();

    Color c1=new Color(50, 105, 168);
    Color c2=new Color(0,0,0);
    Color c3=new Color(255, 255, 255);
    Color c4 =new Color(255, 0, 0);
    Color c5 = new Color(19, 173, 173);

    ChangePassword()
    {
        setSize(400, 300);
        setBounds(300, 200, 400, 300);
```

```
getContentPane().setBackground(c3);

Font f1 = new Font("serif", Font.PLAIN, 18);
Font err_font = new Font("serif", Font.PLAIN, 12);


lbl_currentPass = new JLabel("Current Password :");
lbl_currentPass.setForeground(c2);
lbl_currentPass.setBounds(20, 20, 150, 30);
lbl_currentPass.setFont(f1);
add(lbl_currentPass);

txt_currentPass = new JPasswordField();
txt_currentPass.setBorder(new LineBorder(c2));
txt_currentPass.setBounds(180, 20, 180, 30);
txt_currentPass.addKeyListener(new KeyListener() {
    @Override
    public void keyTyped(KeyEvent e) {
    }

    @Override
    public void keyPressed(KeyEvent e) {
    }

    @Override
    public void keyReleased(KeyEvent e) {
        String phone = new String(txt_currentPass.getPassword());
        if(phone.length() < 1){
            err_current.setText("This Field cannot be empty.");
            add(err_current);
        } else {
            remove(err_current);
        }
        validate();
        repaint();
    }
});
add(txt_currentPass);

err_current = new JLabel();
err_current.setBounds(180, 5, 150, 15);
err_current.setFont(err_font);
err_current.setForeground(Color.red);

JCheckBox check_current = new JCheckBox("Show Password");
check_current.setBackground(c3);
check_current.setForeground(c2);
check_current.setBounds(180, 51, 150, 15);
check_current.setFont(new Font("serif", Font.PLAIN, 13));
check_current.addActionListener(e -> {
    showPassword(check_current, txt_currentPass);
});
add(check_current);


lbl_newPass = new JLabel("New Password :");
lbl_newPass.setForeground(c2);
lbl_newPass.setBounds(20, 80, 150, 30);
lbl_newPass.setFont(f1);
add(lbl_newPass);

txt_newPass = new JPasswordField();
txt_newPass.setBorder(new LineBorder(c2));
txt_newPass.setBounds(180, 80, 180, 30);
txt_newPass.addFocusListener(new FocusListener() {
    @Override
    public void focusGained(FocusEvent e) {
        if(flag){
            showMessage();
            flag = false;
        }
    }

    @Override
    public void focusLost(FocusEvent e) {
        flag = true;
    }
});
txt_newPass.addKeyListener(new KeyListener() {
    @Override
    public void keyTyped(KeyEvent e) {
    }

    @Override
    public void keyPressed(KeyEvent e) {
    }

    @Override
    public void keyReleased(KeyEvent e) {
        String password = new String(txt_newPass.getPassword());
        if(password.length() < 1){
            err_new.setText("This Field cannot be empty");
            add(err_new);
        } else {
            remove(err_new);
        }
        validate();
        repaint();
    }
});
add(txt_newPass);
```

```
err_new = new JLabel();
err_new.setBounds(180, 65, 150, 15);
err_new.setFont(err_font);
err_new.setForeground(Color.red);

JCheckBox check_new = new JCheckBox("Show Password");
check_new.setBackground(c3);
check_new.setForeground(c2);
check_new.setBounds(180, 111, 150, 15);
check_new.setFont(new Font("serif", Font.PLAIN, 13));
check_new.addActionListener(e -> {
   showPassword(check_new, txt_newPass);
});
add(check_new);

lbl_confPass = new JLabel("Confirm Password :");
lbl_confPass.setForeground(c2);
lbl_confPass.setBounds(20,140, 150, 30);
lbl_confPass.setFont(f1);
add(lbl_confPass);

txt_confPass = new JPasswordField();
txt_confPass.setBorder(new LineBorder(c2));
txt_confPass.setBounds(180, 140, 180, 30);
txt_confPass.addKeyListener(new KeyListener() {
   @Override
   public void keyTyped(KeyEvent e) {
   }

   @Override
   public void keyPressed(KeyEvent e) {
   }

   @Override
   public void keyReleased(KeyEvent e) {
      String phone = new String(txt_confPass.getPassword());
      if(phone.length() < 1){
         err_conf.setText("This Field cannot be empty.");
         add(err_conf);
      } else {
         remove(err_conf);
      }
      validate();
      repaint();
   }
});
add(txt_confPass);

err_conf = new JLabel();
err_conf.setBounds(180, 125, 150, 15);
```

```
        err_conf.setFont(err_font);
        err_conf.setForeground(Color.red);

        JCheckBox check_conf = new JCheckBox("Show Password");
        check_conf.setBackground(c3);
        check_conf.setForeground(c2);
        check_conf.setBounds(180, 171, 150, 15);
        check_conf.setFont(new Font("serif", Font.PLAIN, 13));
        check_conf.addActionListener(e -> {
            showPassword(check_conf, txt_confPass);
        });
        add(check_conf);

        JButton btn_change = new JButton("Change");
        btn_change.setBorder(new LineBorder(c2));
        btn_change.setBackground(c1);
        btn_change.setForeground(c3);
        btn_change.setBounds(200, 200, 100, 30);
        btn_change.addActionListener(e -> {
            handleChange();
        });
        add(btn_change);

        JButton btn_cancel = new JButton("Cancel");
        btn_cancel.setBorder(new LineBorder(c2));
        btn_cancel.setBackground(c1);
        btn_cancel.setForeground(c3);
        btn_cancel.setBounds(80, 200, 100, 30);
        btn_cancel.addActionListener(e -> {
            dispose();
        });
        add(btn_cancel);

        setResizable(false);
        setLayout(null);
        setVisible(true);

    }

    void showMessage(){
        UI.put("OptionPane.background", c3);
        UI.put("Panel.background", c3);
        JOptionPane.showMessageDialog(this, "Your Password must contain a lower case and 
uppercase alphabet , a number and must be minimum 8 digits long.");
    }

    void showPassword(JCheckBox check, JPasswordField pass) {
        if(check.isSelected())
            pass.setEchoChar('\u0000');
        else
            pass.setEchoChar((Character)UIManager.get("PasswordField.echoChar"));
```

```
}
void handleChange(){
    currentPass = new String(txt_currentPass.getPassword());
    newPass = new String(txt_newPass.getPassword());
    confPass = new String(txt_confPass.getPassword());
    boolean flag = true;

    if(currentPass.length() < 1){
        err_current.setText("This Field cannot be empty.");
        add(err_current);
        flag = false;
    }
    else if(!checkCurrentPassword()){
        err_current.setText("Invalid current password.");
        add(err_current);
        flag = false;
    }

    if(newPass.length() < 1){
        err_new.setText("This Field cannot be empty.");
        add(err_new);
        flag = false;
    } else if (!Pattern.matches("^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z]).{8,32}$", newPass)){
        txt_newPass.setText("");
        txt_confPass.setText("");
        err_new.setText("Password must follow the rules");
        add(err_new);
        flag = false;
    } else if(!newPass.equals(confPass)) {
        err_new.setText("The passwords must match.");
        add(err_new);
        err_conf.setText("The passwords must match.");
        add(err_conf);
        txt_newPass.setText("");
        txt_confPass.setText("");
        flag = false;
    }

    if(confPass.length() < 1){
        err_conf.setText("This Field cannot be empty.");
        add(err_conf);
        flag = false;
    }

    validate();
    repaint();

    if(flag) {
        changePassword();
    }
    }

    boolean checkCurrentPassword(){
        try {
            Connection con = DbConnection.getConnection();
            Statement stmt = con.createStatement();
            String query;
            if(UserType.userType.equals("Public")){
                query = "select password from public where id="+PublicModel.id;
            }
            else {
                query = "select password from govt_servant where id=" + GovtModel.id;
            }

            ResultSet user = stmt.executeQuery(query);
            user.next();
            String pass = user.getString("password");
            String hashedPassword = HashPassword.toHexString(currentPass);
            return pass.equals(hashedPassword);

        } catch(SQLException ex) {
            System.out.println("Exception occurred while checking current password " + ex);
        }
        return false;
    }

    void changePassword() {
        try {
            Connection con = DbConnection.getConnection();
            Statement stmt = con.createStatement();
            String hashedPassword = HashPassword.toHexString(newPass);
            String query;
            if(UserType.userType.equals("Public")){
                query = "update public set password='"+hashedPassword+"' where id="+PublicModel.id;
            }
            else{
                query = "update govt_servant set password='"+hashedPassword+"' where
id="+GovtModel.id;
            }
            int result = stmt.executeUpdate(query);
            JOptionPane.showMessageDialog(this, "Password succcessfully updated.");
            UI.put("OptionPane.background", c3);
            UI.put("Panel.background", c3);
            dispose();
            con.close();
        } catch(SQLException ex) {
            System.out.println("Excepton occurred while changing password  "+ex);

        }
    }
}
```

## HASH PASSWORD:

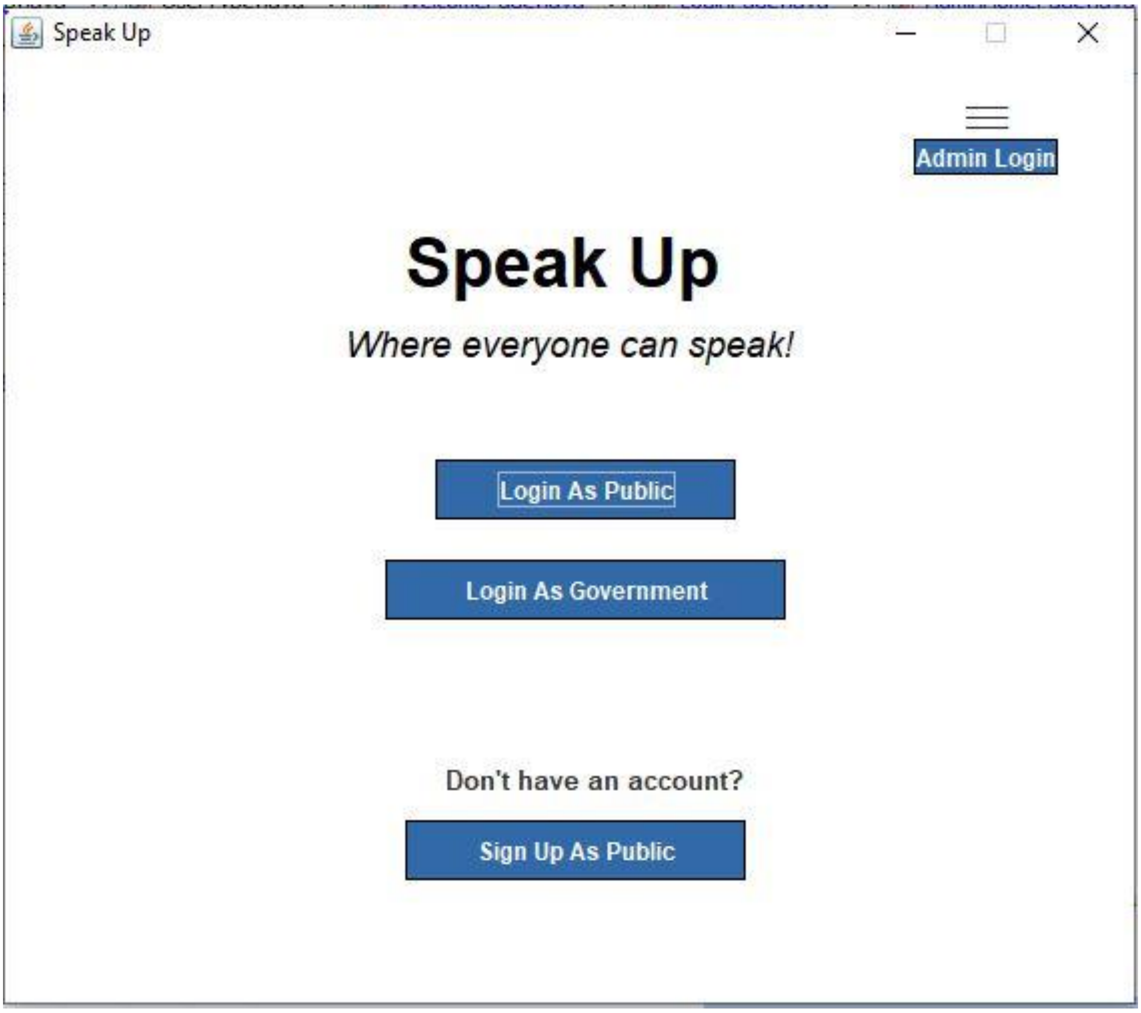
```
package speak_up;

import java.math.BigInteger;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;



public class HashPassword {
   public static byte[] getSHA(String input) throws NoSuchAlgorithmException {
      MessageDigest md = MessageDigest.getInstance("SHA-256");
      return md.digest(input.getBytes(StandardCharsets.UTF_8));
   }

   public static String toHexString(String password){
      try{
         byte[] hash = getSHA(password);

         BigInteger number = new BigInteger(1, hash);
         StringBuilder hexString = new StringBuilder(number.toString(16));
         while (hexString.length() < 32){
            hexString.insert(0, '0');
         }
         return hexString.toString();
      } catch(NoSuchAlgorithmException ex) {
         System.out.println("Exception occurred while hashing password");
      }
      return null;
   }
}
```

## DATABASE CONNECTION:

```
package speak_up;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DbConnection {

    static Connection getConnection() {
        try {
            return DriverManager.getConnection("jdbc:postgresql://127.0.0.1:5432/speak_up_1", "postgres", "23@2000rAGHU");
        } catch(SQLException ex) {
            System.out.println("Error establishing conection to DB "+ ex);
        }
        return null;
    }
}
```

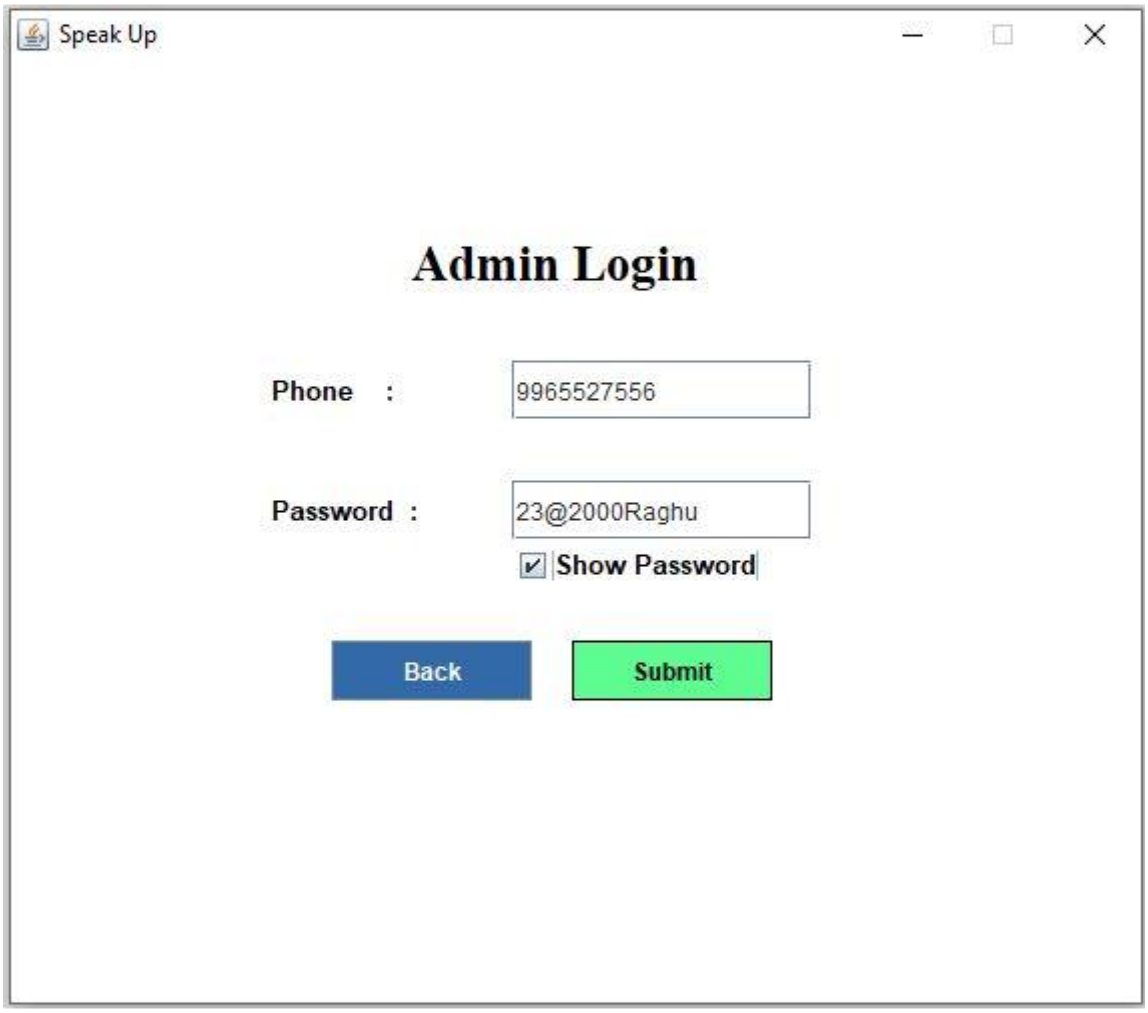
## EXECUTION SCREENSHOT:

## WELCOME PAGE:



## ADMIN MODULE →

## ADMIN LOGIN PAGE:



Speak Up

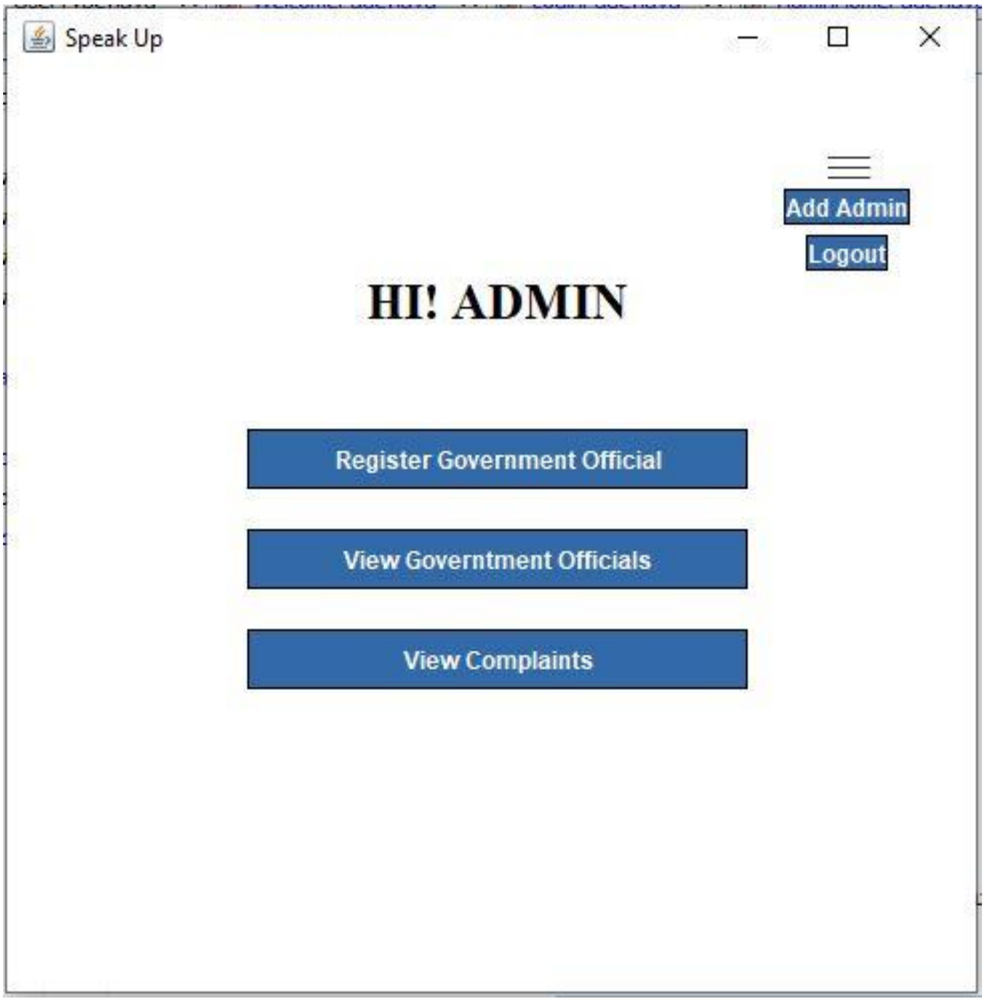
Admin Login

Phone : 9965527556

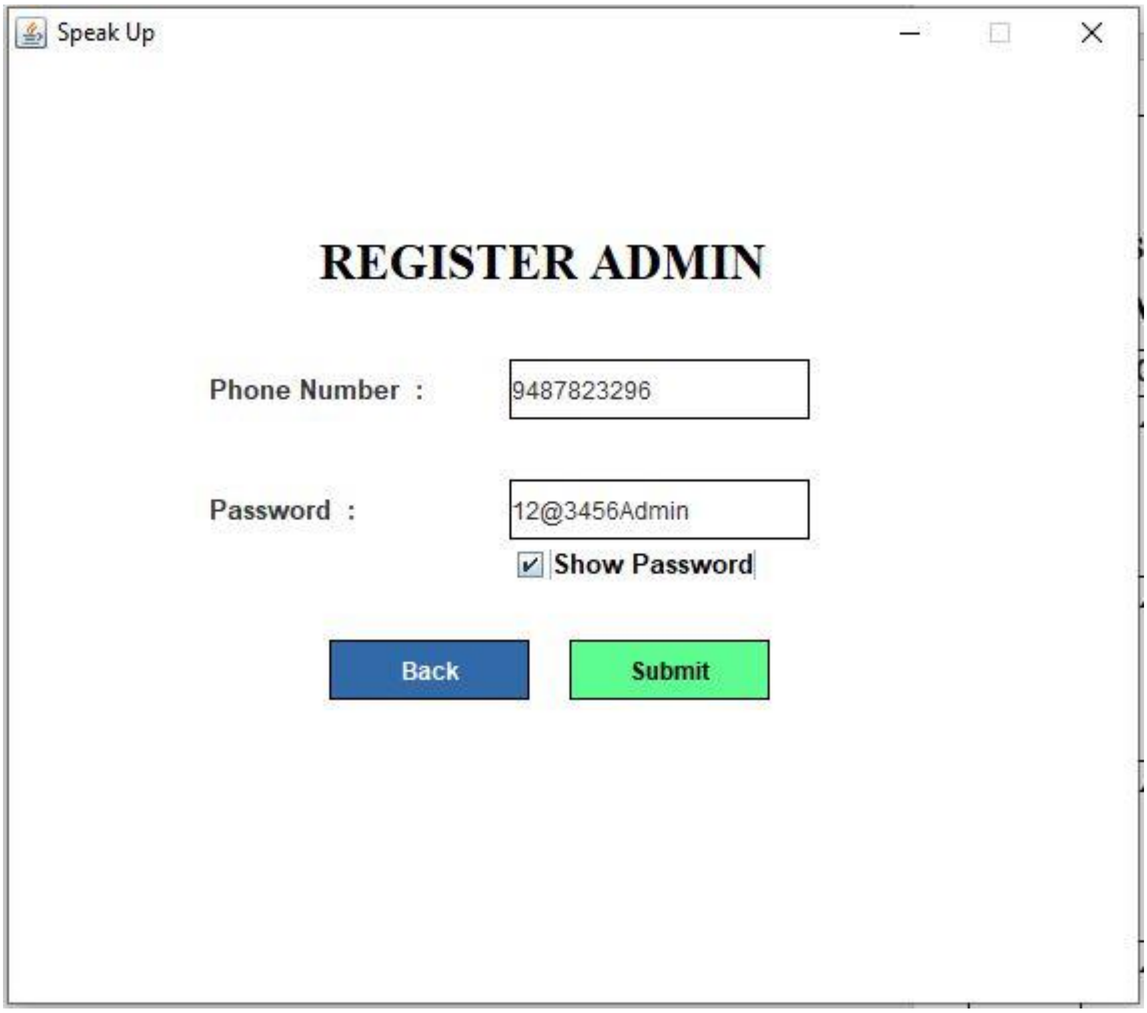
Password : 23@2000Raghu

☑ Show Password

## HOME PAGE:


Speak Up
Add Admin
Logout
HI! ADMIN
Register Government Official
View Governtment Officials
View Complaints

## REGISTER ADMIN:



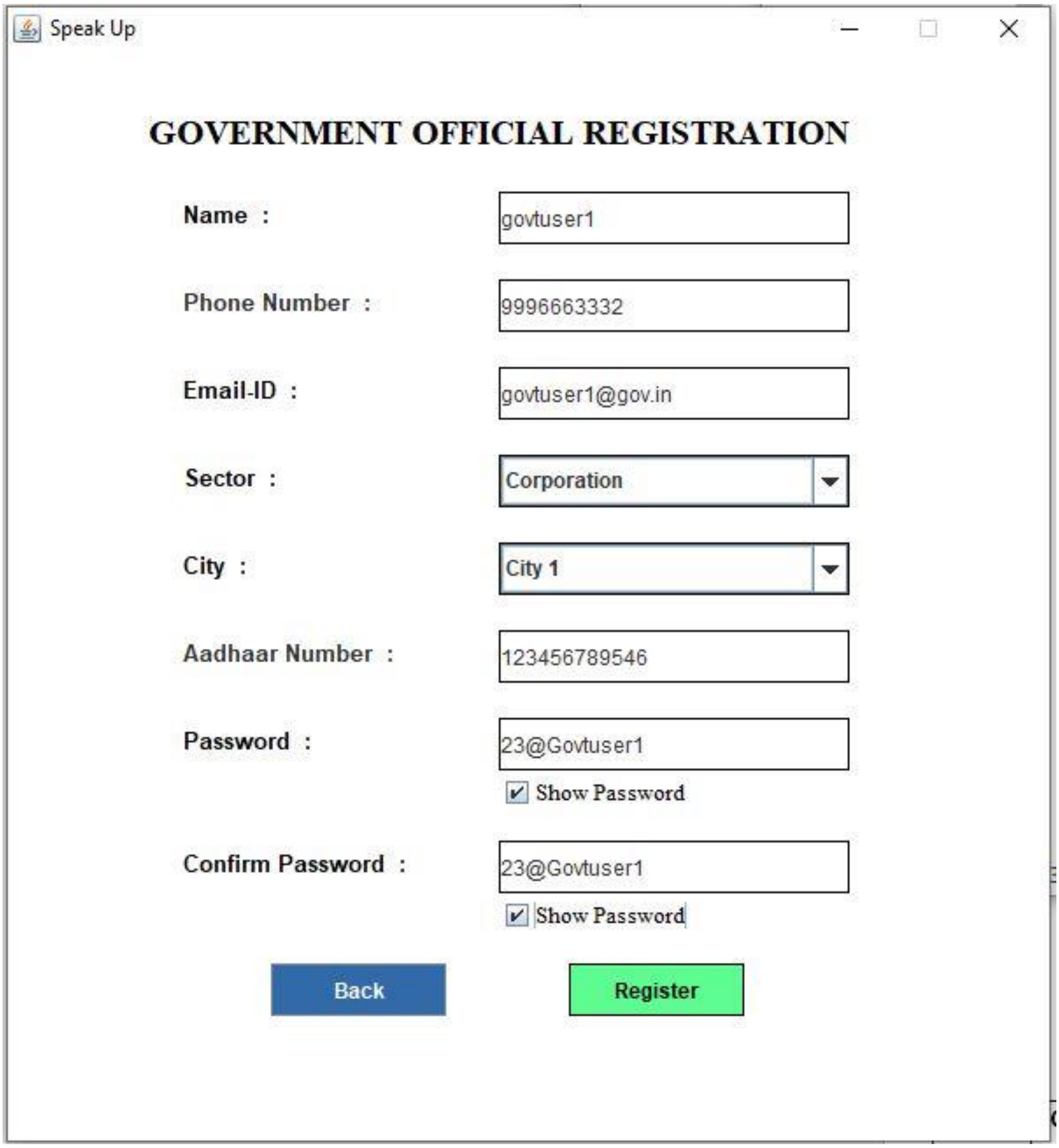
REGISTER ADMIN

Phone Number : 9487823296

Password : 12@3456Admin

☑ Show Password

## REGISTER GOVERNMENT OFFICIAL:



GOVERNMENT OFFICIAL REGISTRATION

Name : govtuser1

Phone Number : 9996663332

Email-ID : govtuser1@gov.in

Sector : Corporation

City : City 1

Aadhaar Number : 123456789546

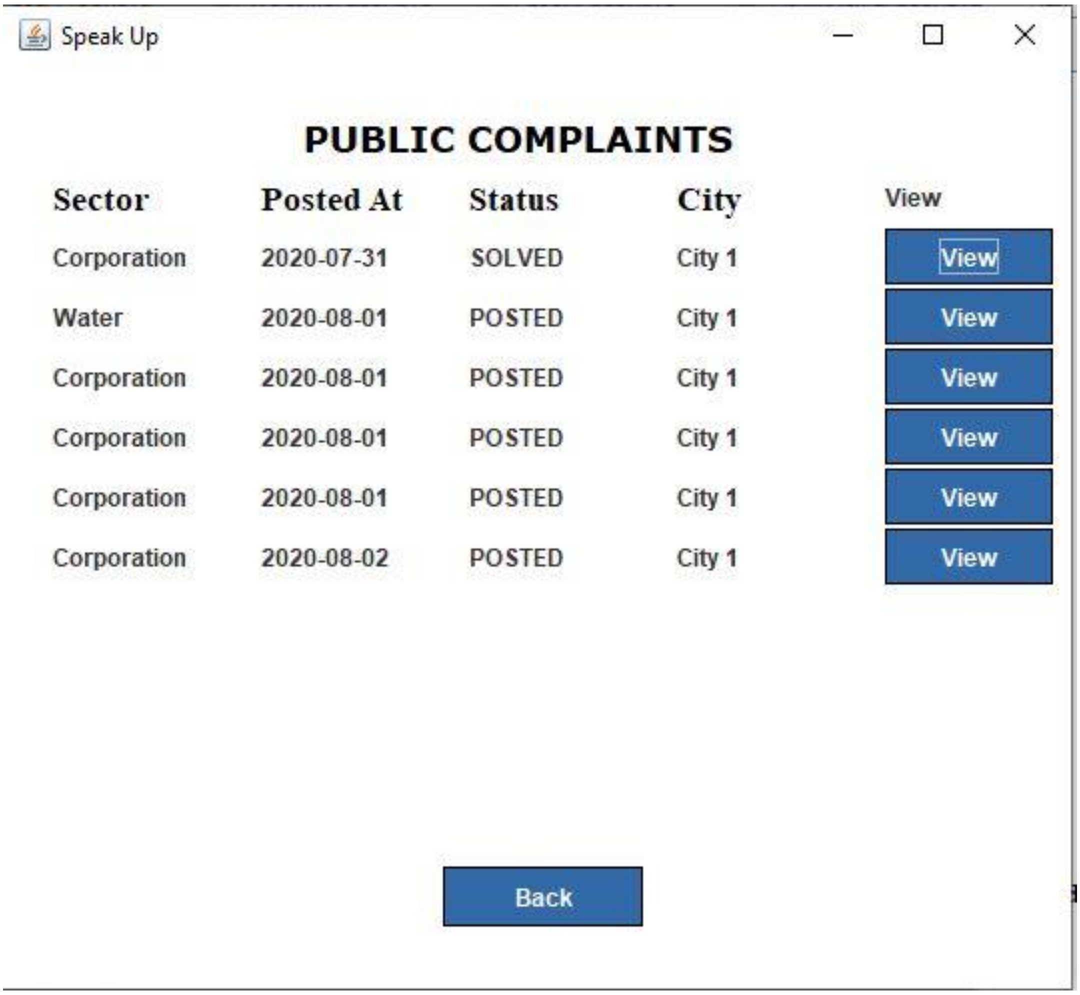
Password : 23@Govtuser1

☑ Show Password

Confirm Password : 23@Govtuser1

☑ Show Password

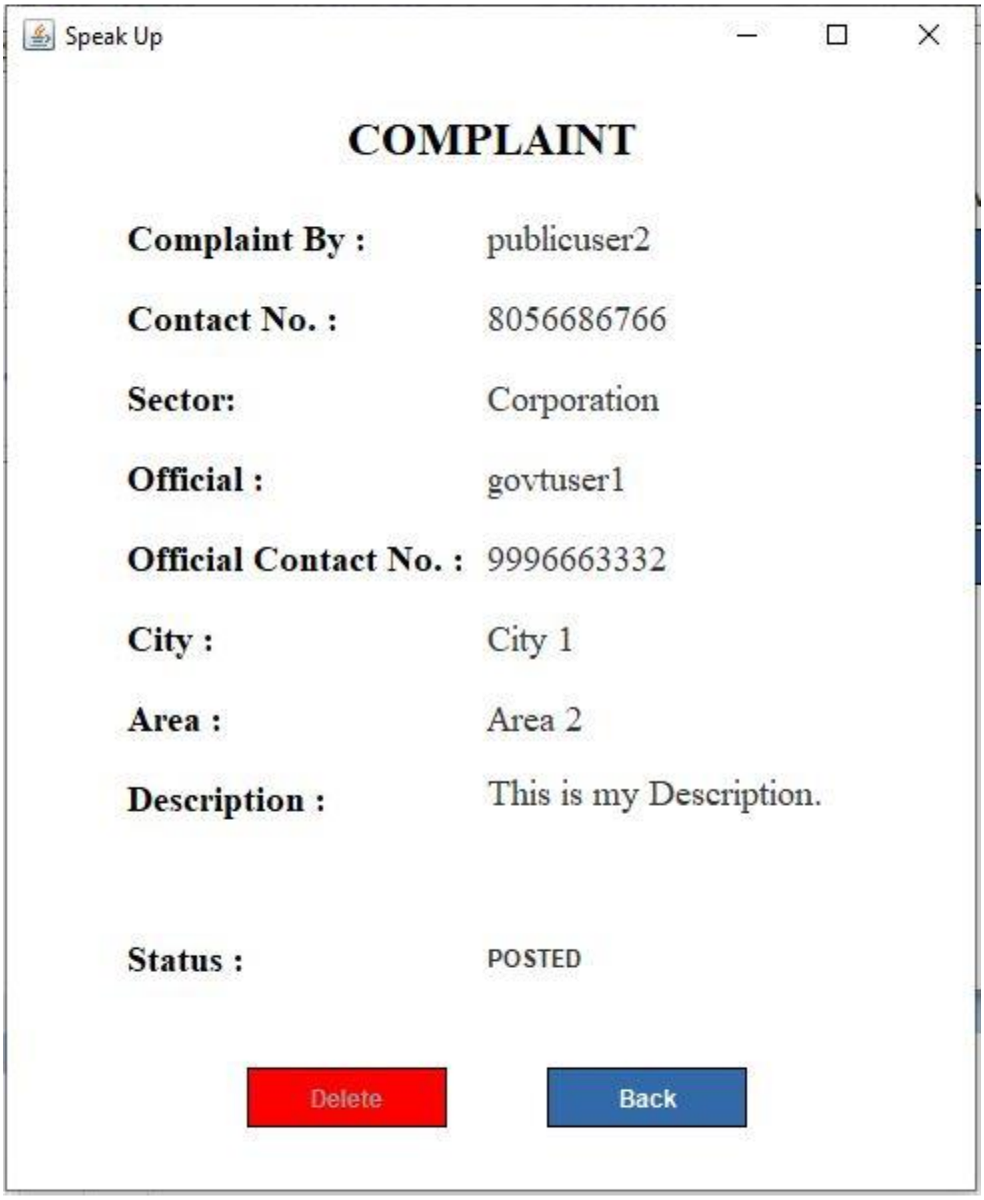
## VIEW COMPLAINTS:



PUBLIC COMPLAINTS

| Sector | Posted At | Status | City | View |
|---|---|---|---|---|
| Corporation | 2020-07-31 | SOLVED | City 1 | View |
| Water | 2020-08-01 | POSTED | City 1 | View |
| Corporation | 2020-08-01 | POSTED | City 1 | View |
| Corporation | 2020-08-01 | POSTED | City 1 | View |
| Corporation | 2020-08-01 | POSTED | City 1 | View |
| Corporation | 2020-08-02 | POSTED | City 1 | View |

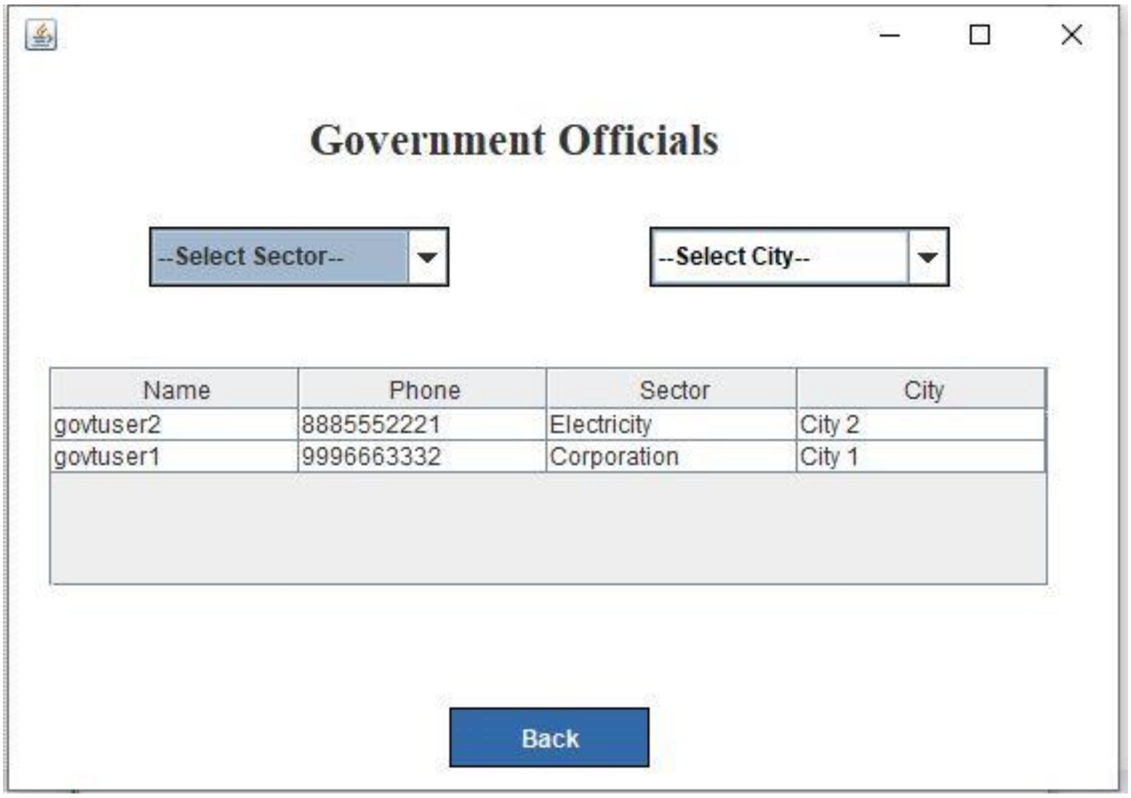
## VIEW DETAILED COMPLAINT:



COMPLAINT

| | |
|---|---|
| Complaint By : | publicuser2 |
| Contact No. : | 8056686766 |
| Sector: | Corporation |
| Official : | govtuser1 |
| Official Contact No. : | 9996663332 |
| City : | City 1 |
| Area : | Area 2 |
| Description : | This is my Description. |
| Status : | POSTED |

## VIEW GOVERNMENT OFFICIALS:

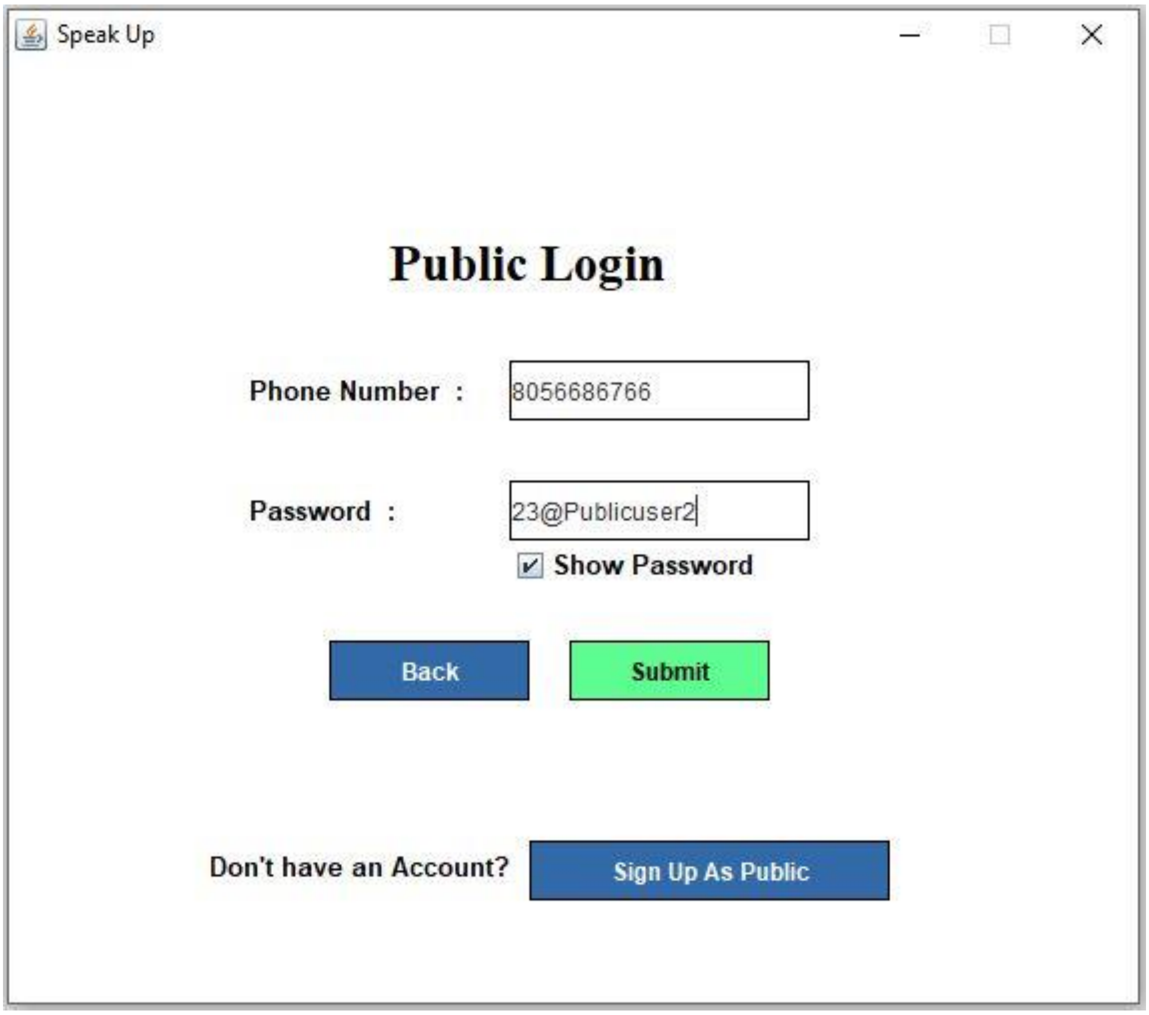


Government Officials

| Name | Phone | Sector | City |
|---|---|---|---|
| govtuser2 | 8885552221 | Electricity | City 2 |
| govtuser1 | 9996663332 | Corporation | City 1 |

## PUBLIC MODULE →

## PUBLIC LOGIN PAGE:



Speak Up

Public Login

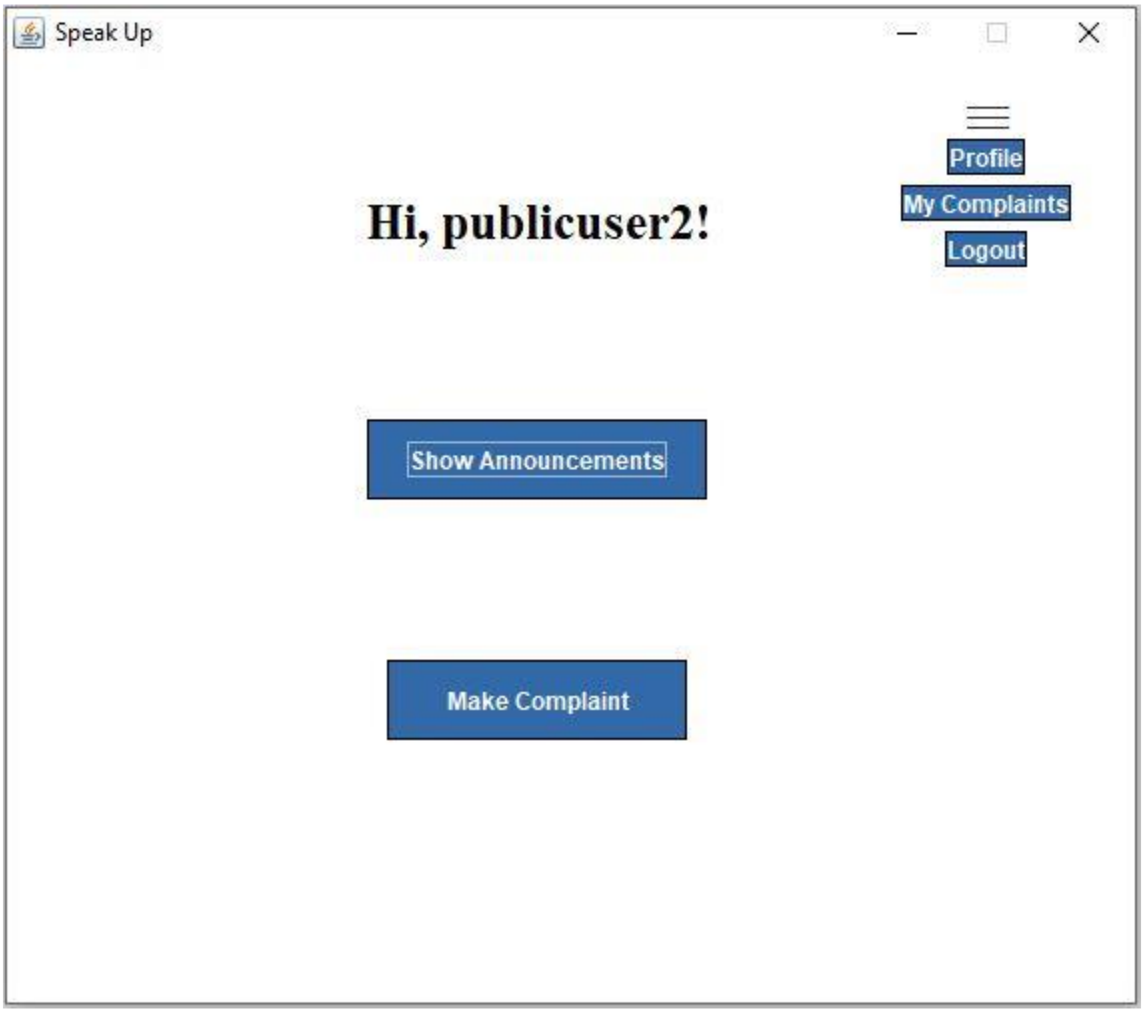
Phone Number : 8056686766

Password : 23@Publicuser2

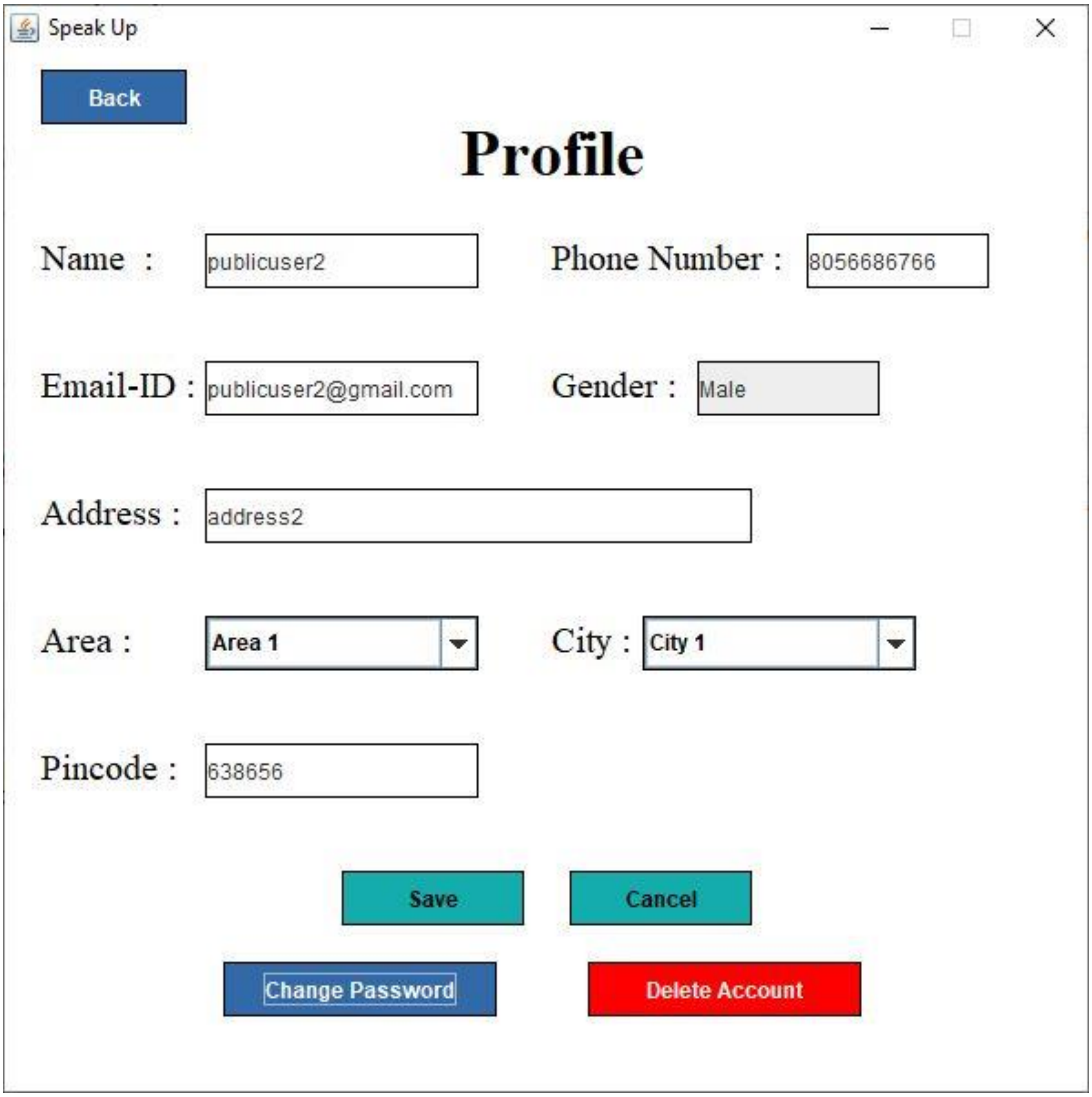
☑ Show Password

Don't have an Account?

## HOME PAGE:

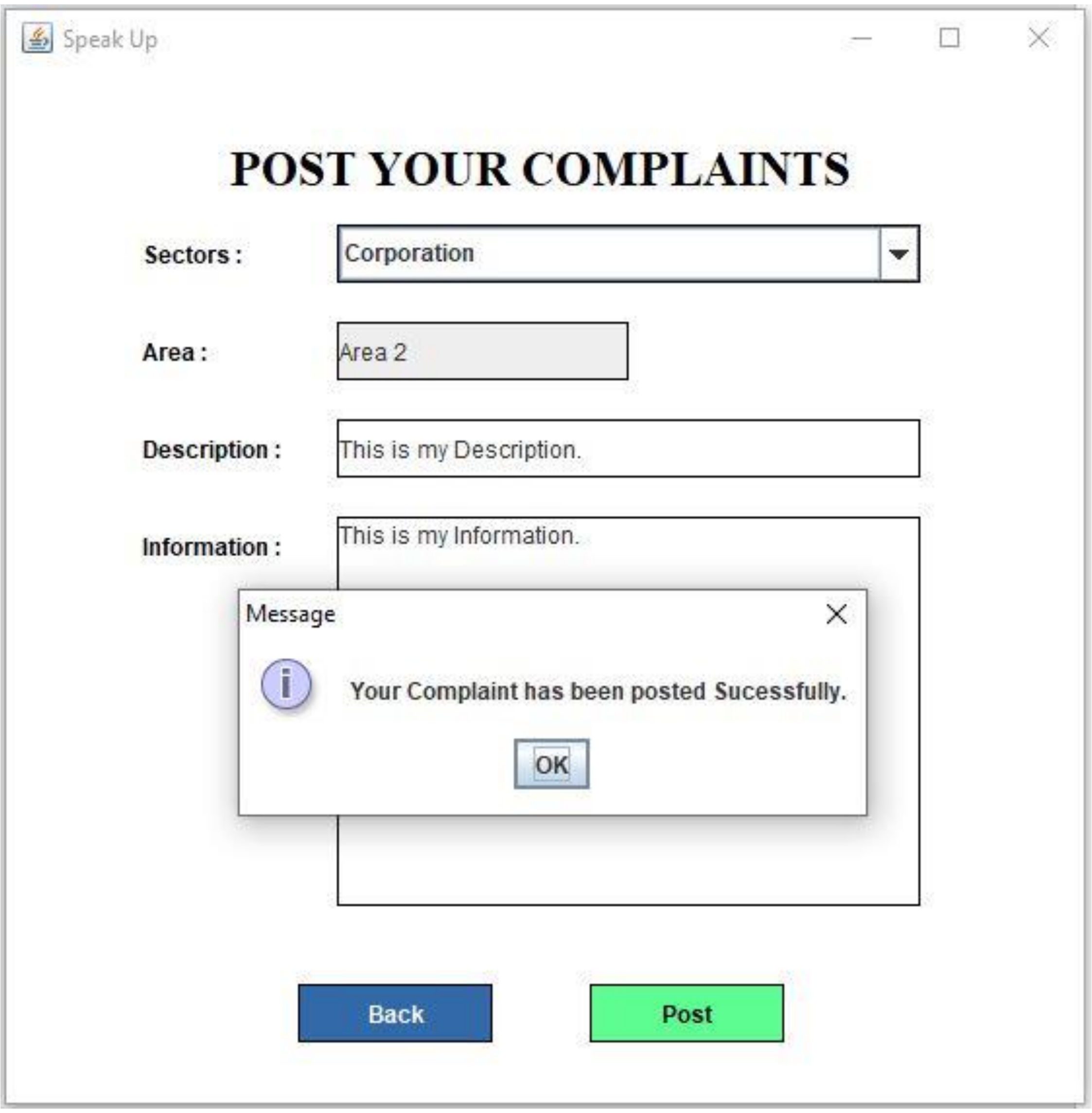

Speak Up
Profile
My Complaints
Logout
Hi, publicuser2!
Show Announcements
Make Complaint

**PROFILE PAGE:**



| | | | |
|---|---|---|---|
| Name : | publicuser2 | Phone Number : | 8056686766 |
| Email-ID : | publicuser2@gmail.com | Gender : | Male |
| Address : | address2 | | |
| Area : | Area 1 | City : | City 1 |
| Pincode : | 638656 | | |

## MAKE COMPLAINT PAGE:



Speak Up

POST YOUR COMPLAINTS

Sectors : Corporation

Area : Area 2

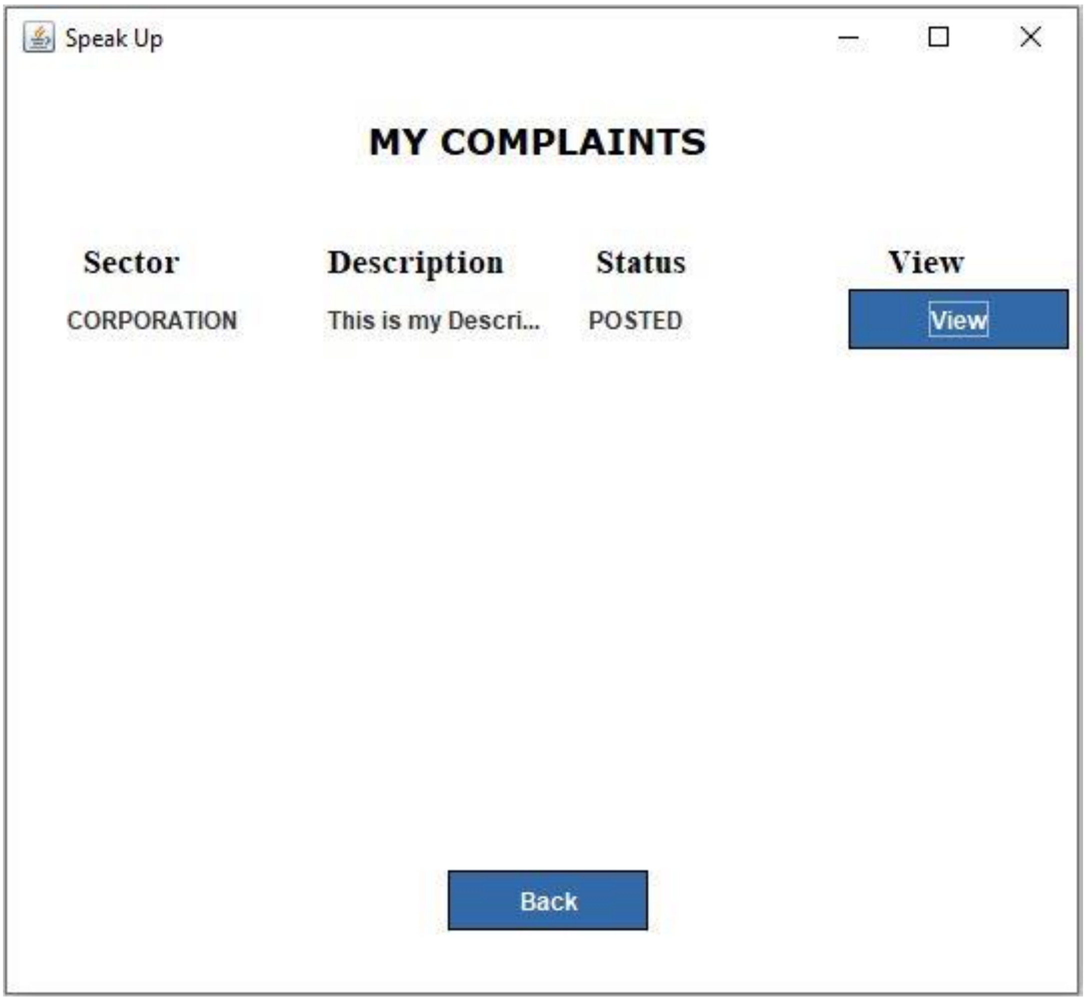
Description : This is my Description.

Information : This is my Information.

Message

Your Complaint has been posted Sucessfully.

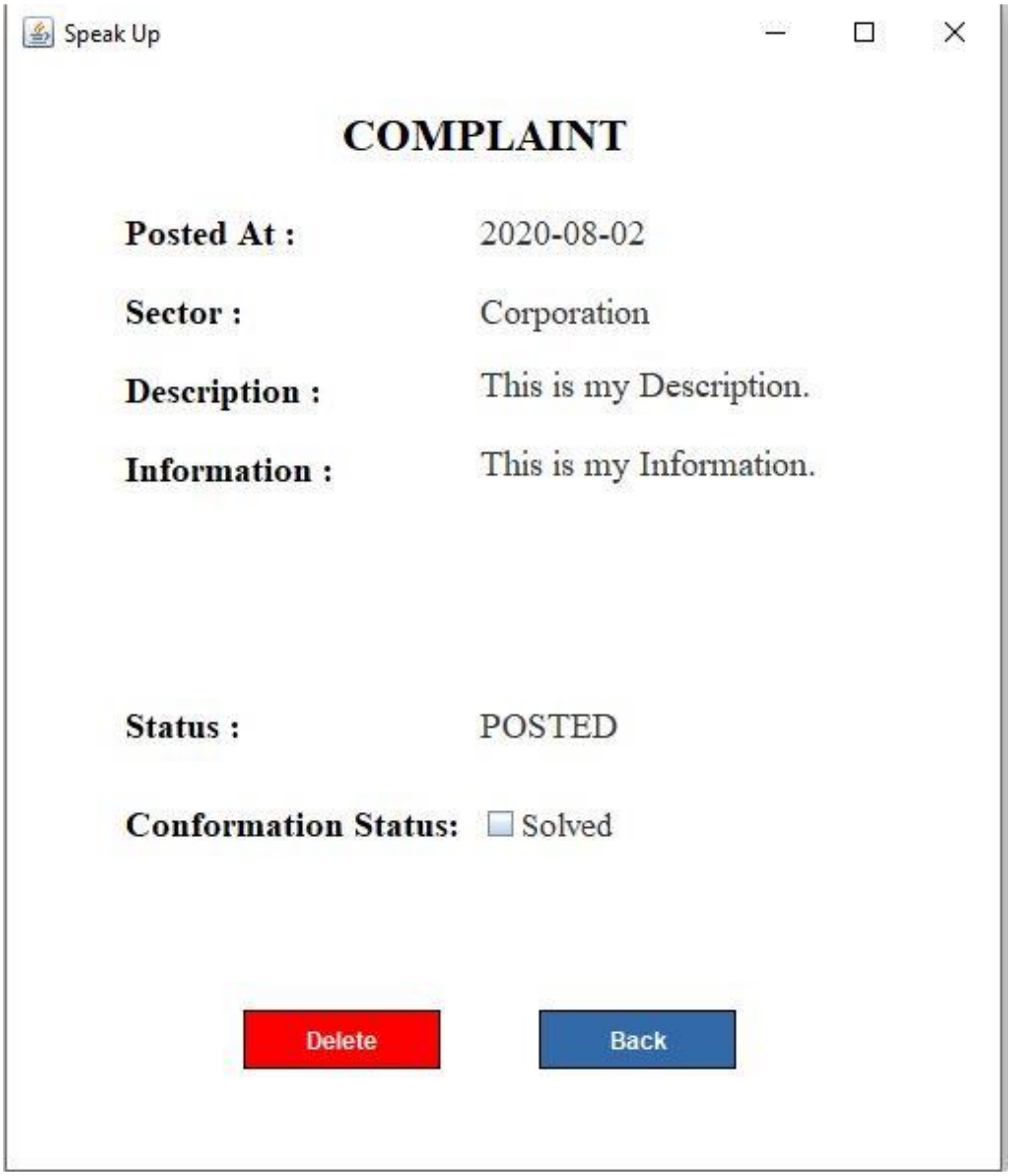
## MY COMPLAINTS PAGE:



MY COMPLAINTS

| Sector | Description | Status | View |
|---|---|---|---|
| CORPORATION | This is my Descri... | POSTED | View |

## VIEW DETAILED COMPLAINT PAGE:



COMPLAINT

Posted At : 2020-08-02

Sector : Corporation

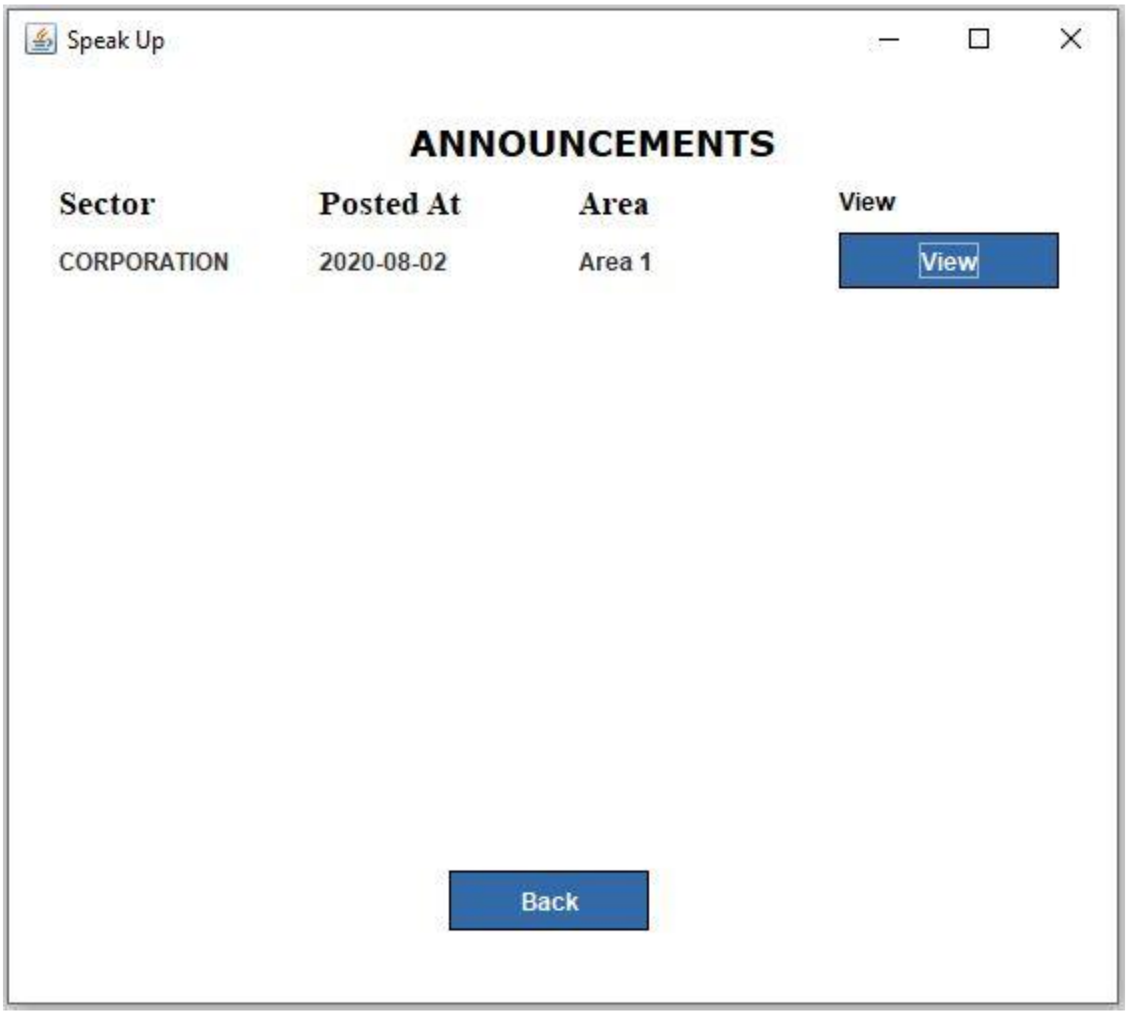
Description : This is my Description.

Information : This is my Information.

Status : POSTED

Conformation Status: □ Solved

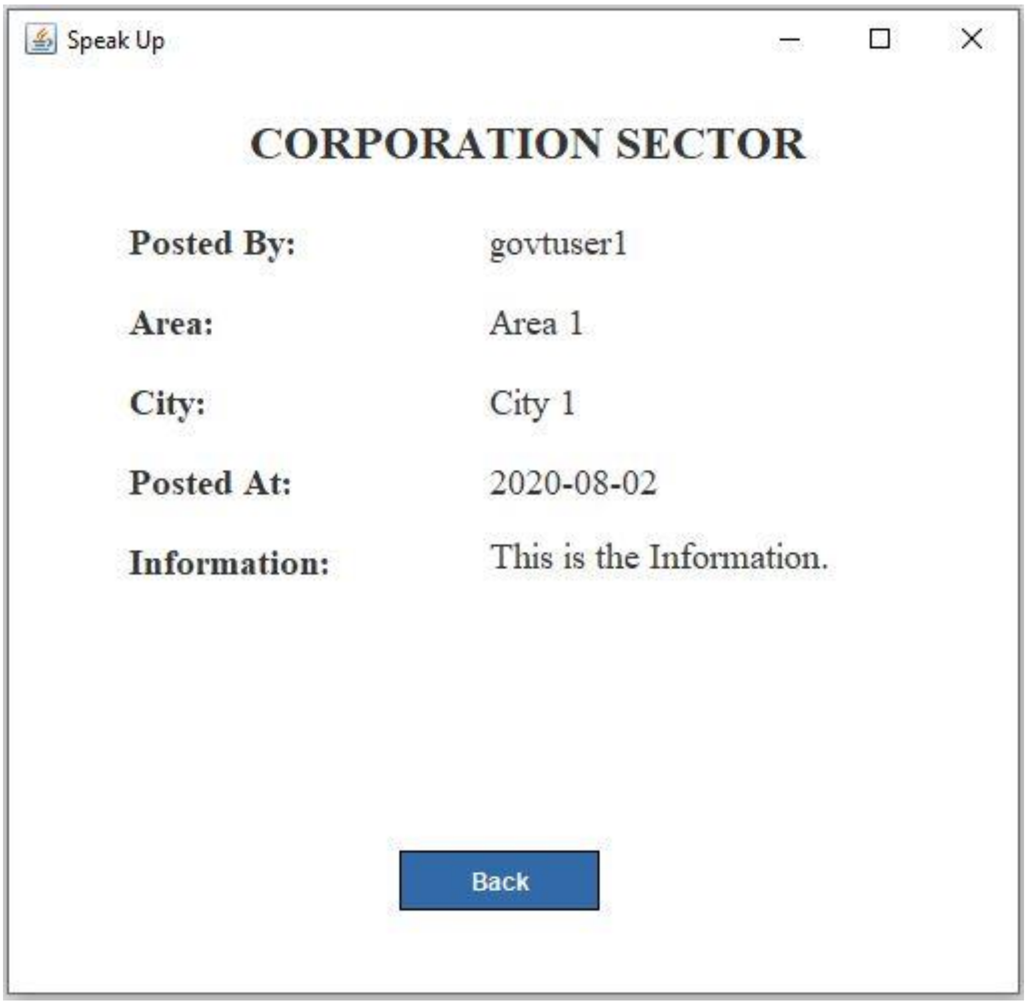
## VIEW ANNOUNCEMENT PAGE:



ANNOUNCEMENTS

| Sector | Posted At | Area | View |
|---|---|---|---|
| CORPORATION | 2020-08-02 | Area 1 | View |

## VIEW DETAILED ANNOUNCEMENT PAGE:

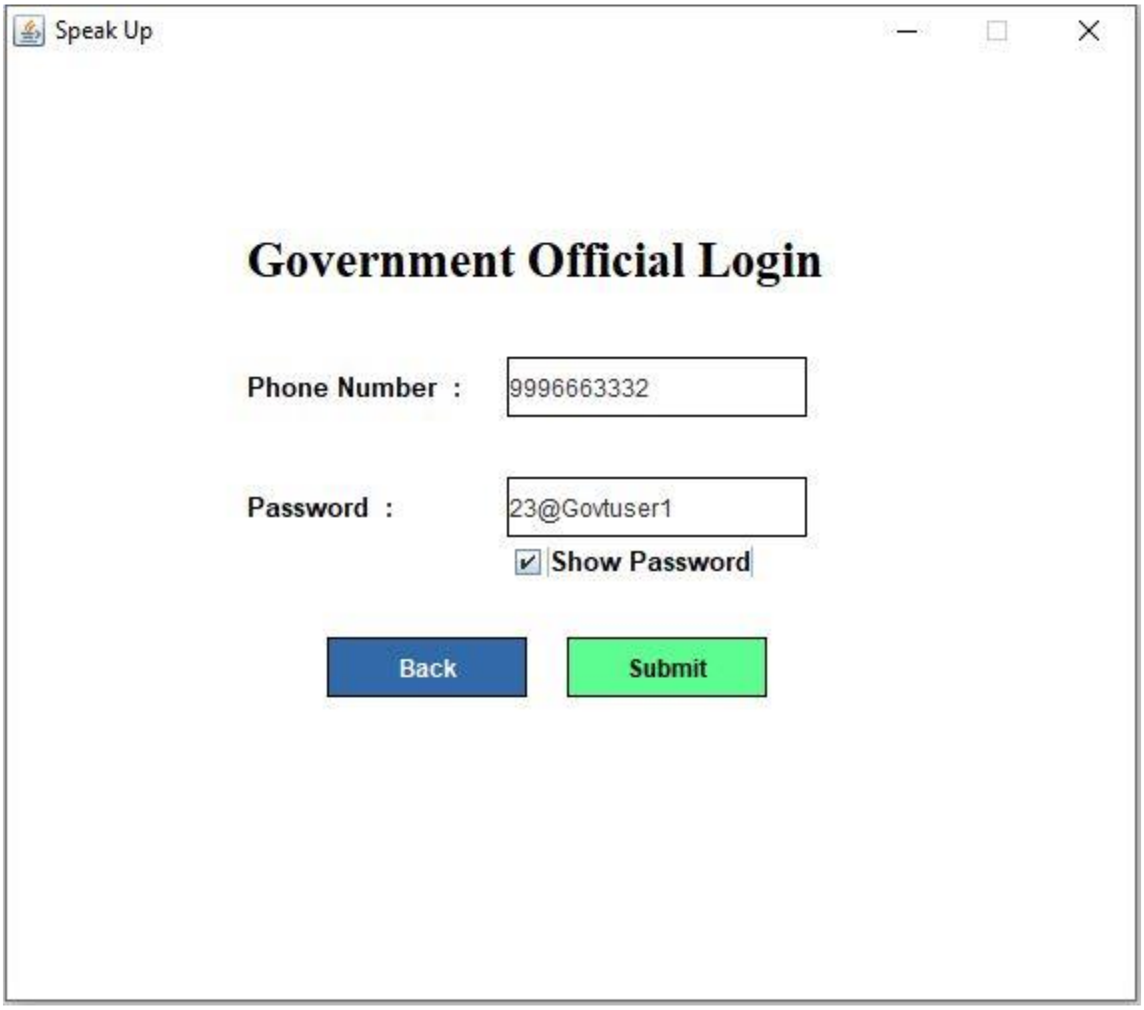


CORPORATION SECTOR

| | |
|---|---|
| **Posted By:** | govtuser1 |
| **Area:** | Area 1 |
| **City:** | City 1 |
| **Posted At:** | 2020-08-02 |
| **Information:** | This is the Information. |

## GOVERNMENT OFFICIAL MODULE →

## GOVERNMENT OFFICIAL LOGIN PAGE:



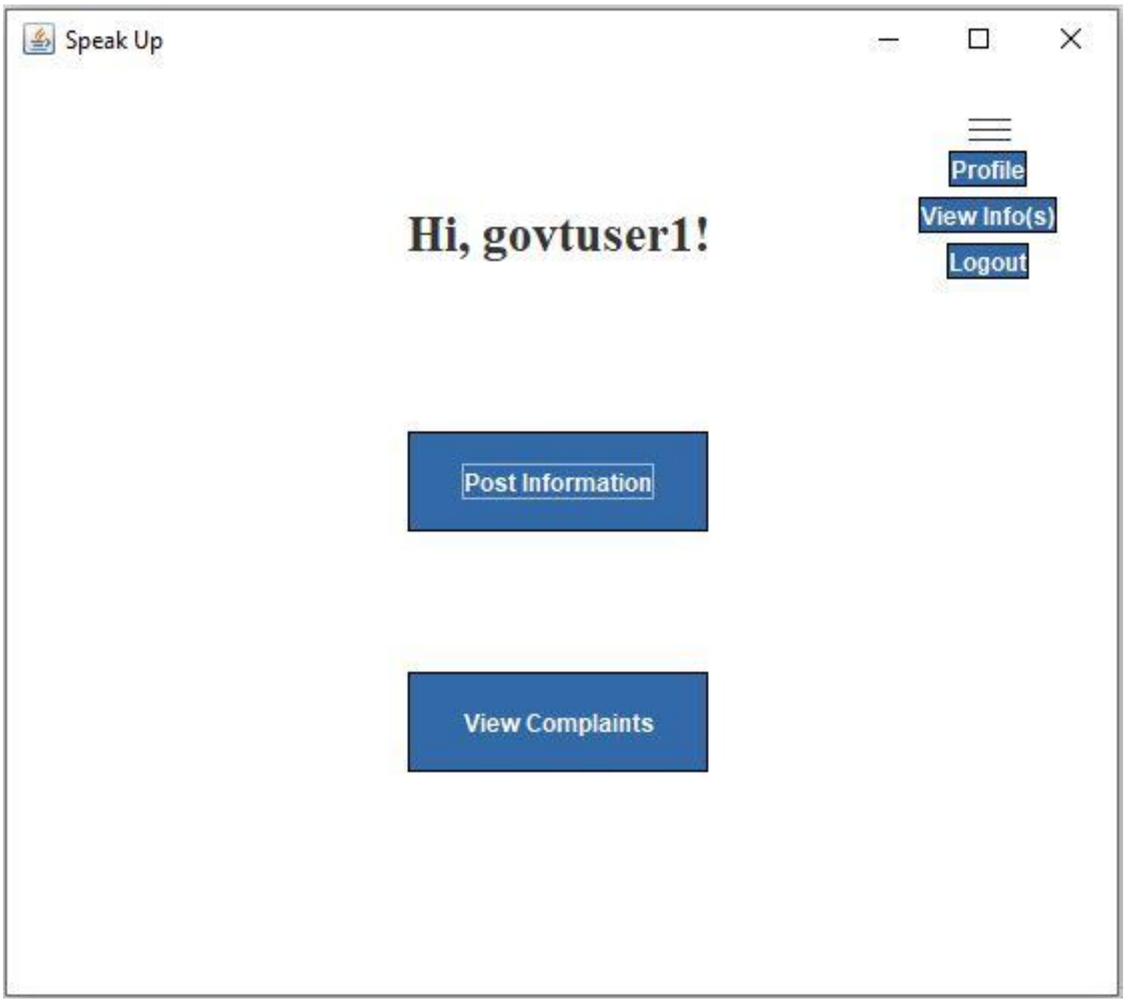
Government Official Login

Phone Number : 9996663332

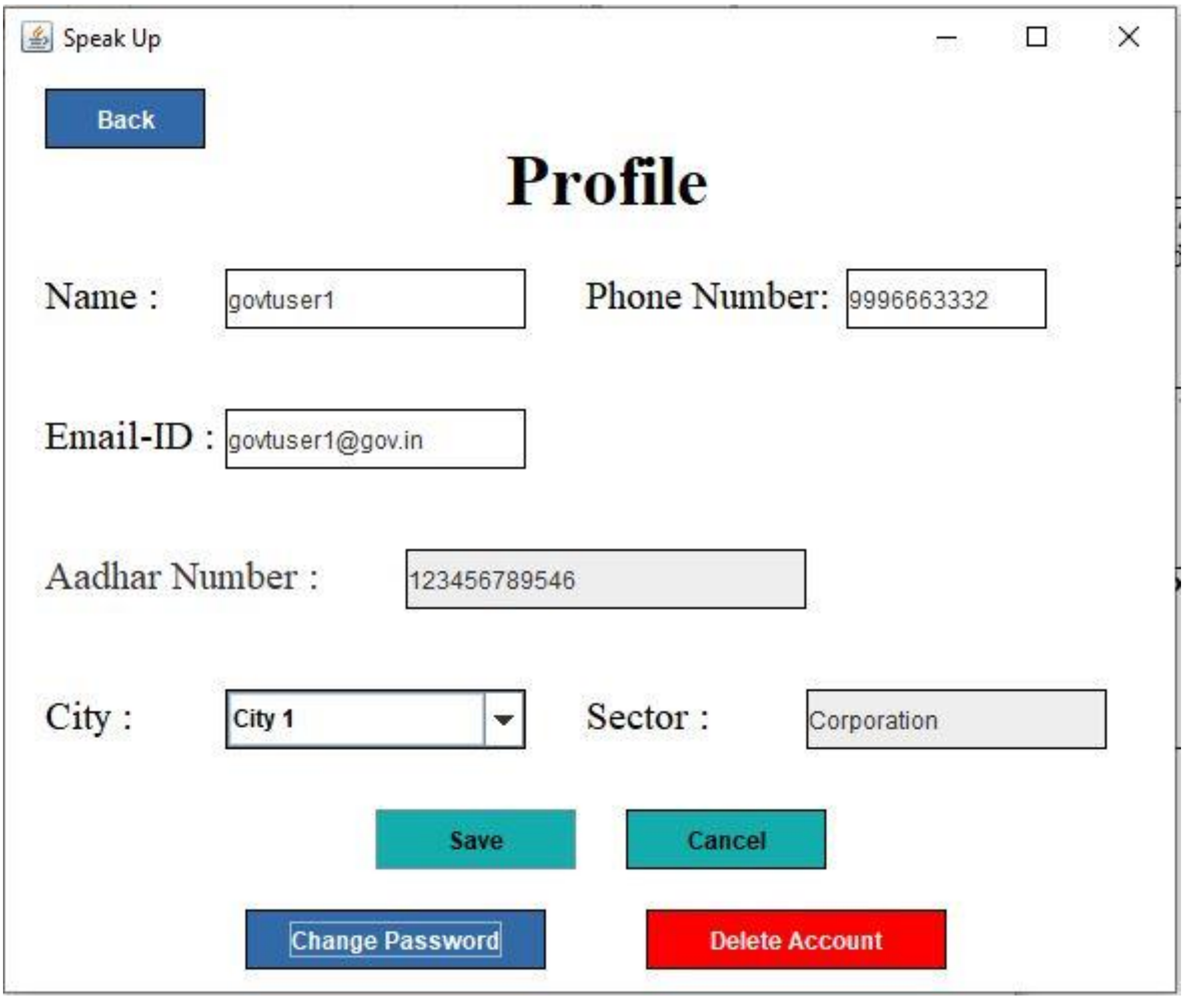
Password : 23@Govtuser1

☑ Show Password

## HOME PAGE:


Speak Up
Profile
View Info(s)
Logout
Hi, govtuser1!
Post Information
View Complaints

## PROFILE PAGE:



Profile

Name : govtuser1

Phone Number: 9996663332

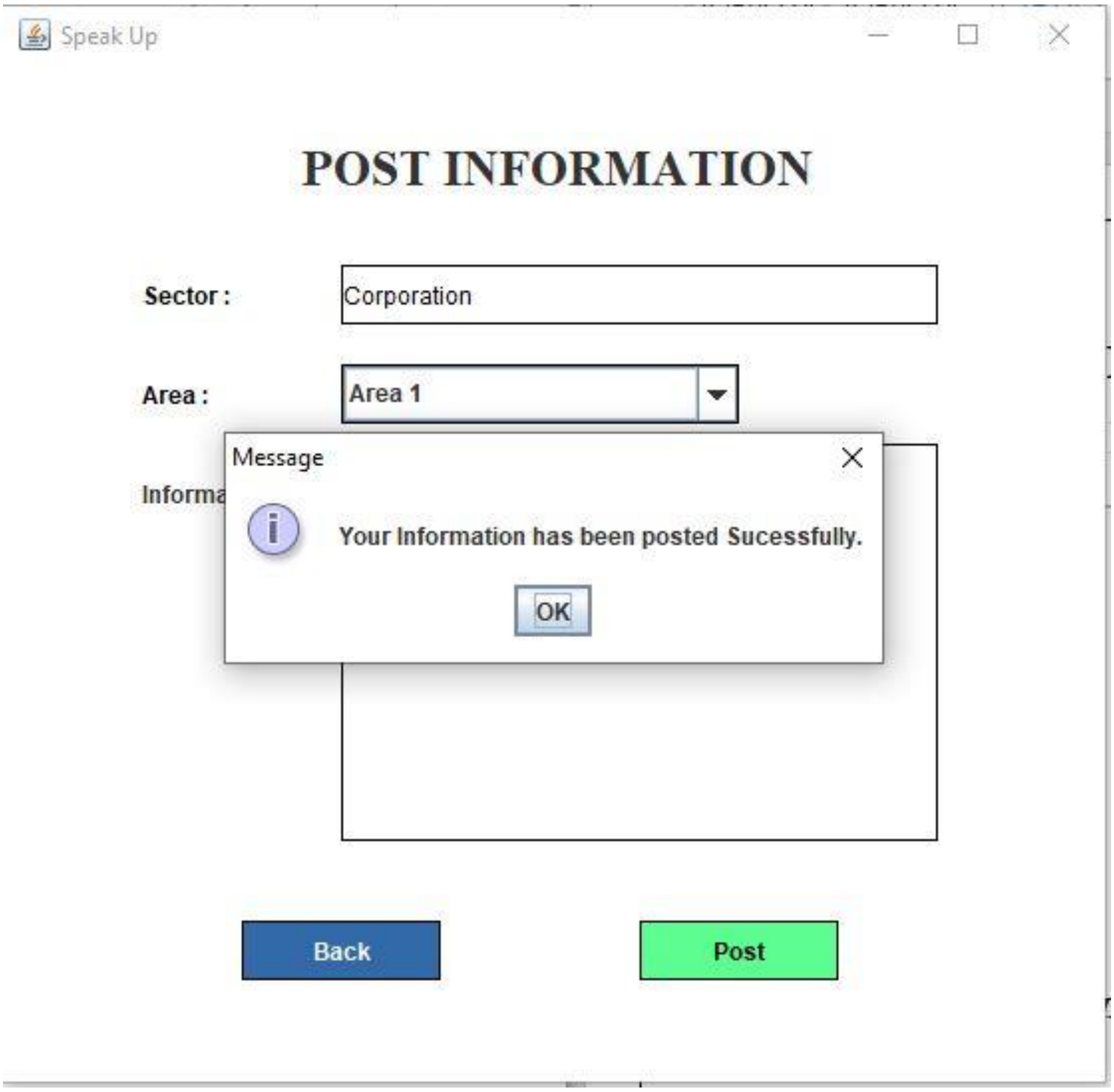
Email-ID : govtuser1@gov.in

Aadhar Number : 123456789546

City : City 1

Sector : Corporation

## POST INFORMATION PAGE:



POST INFORMATION

Sector : Corporation

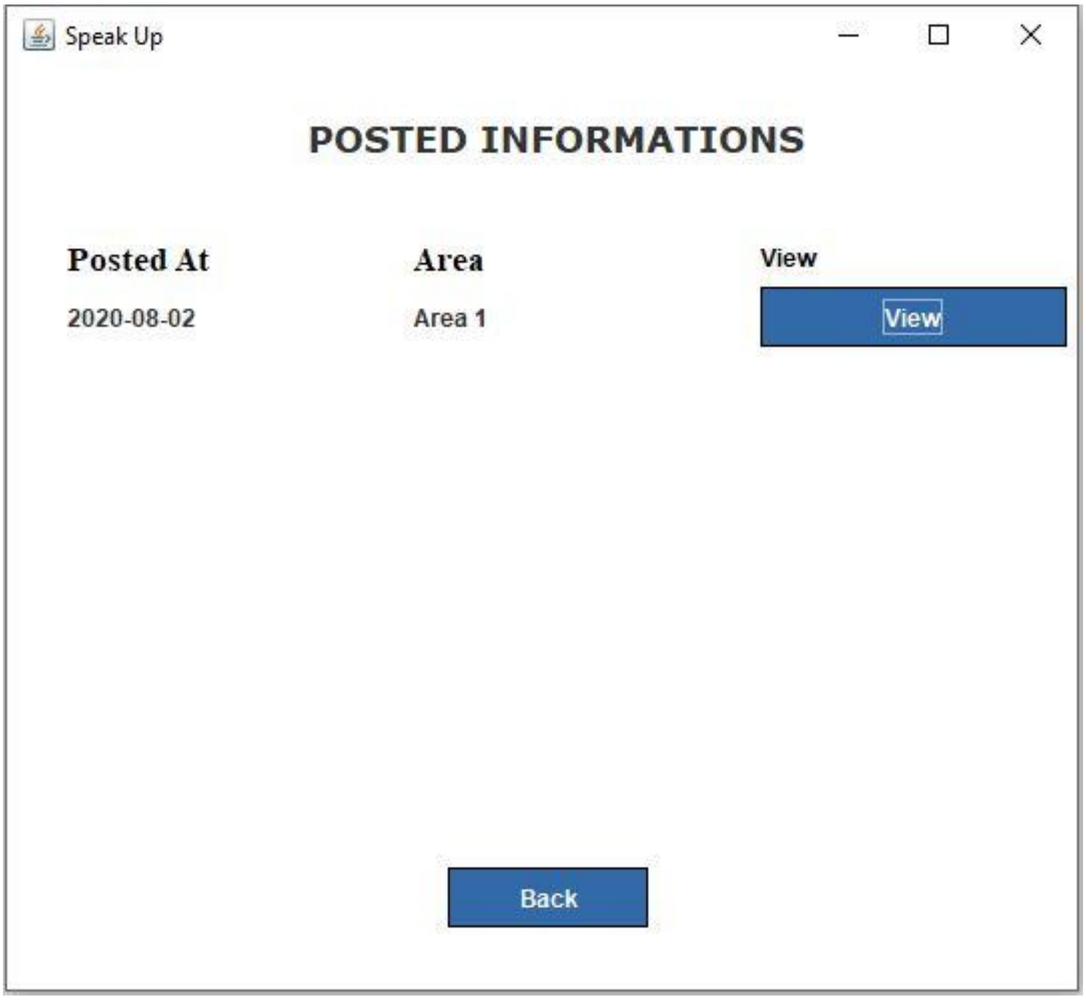
Area : Area 1

Informa

Message

Your Information has been posted Sucessfully.

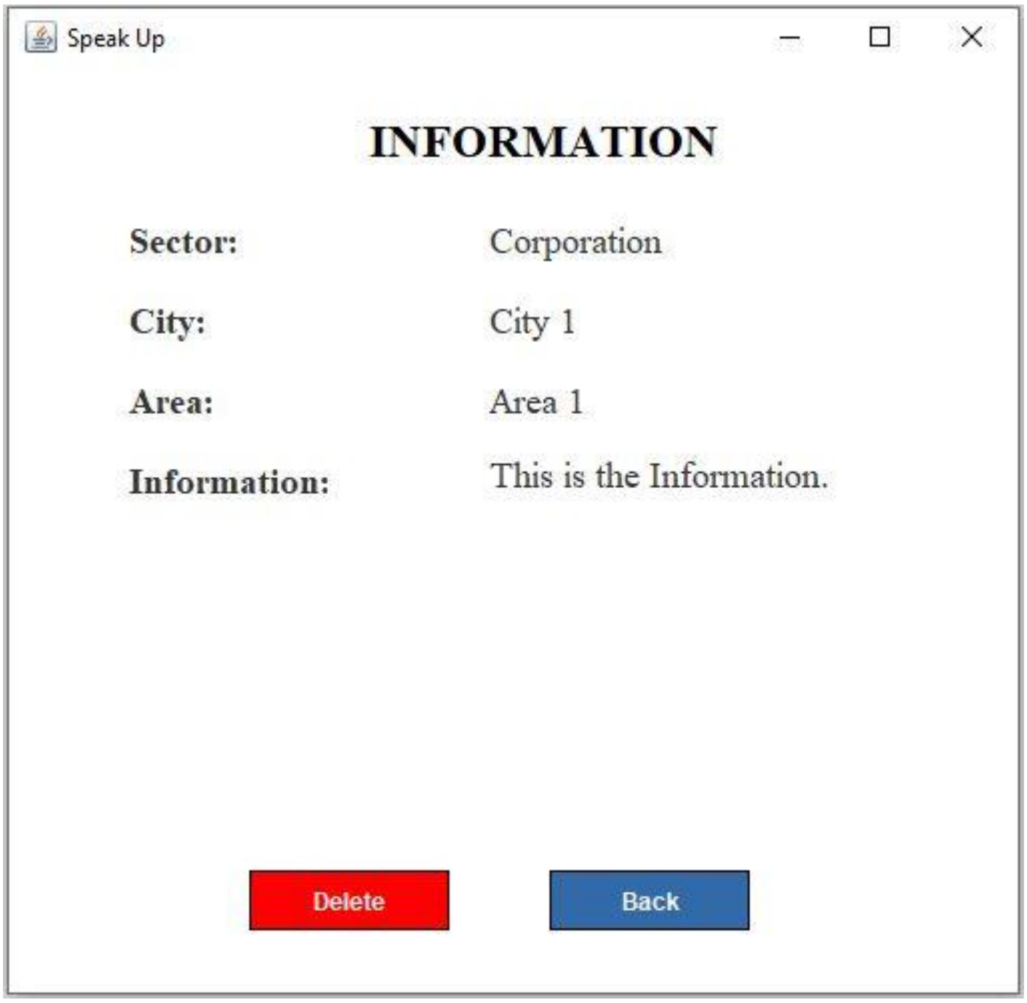
## MY INFO(S) PAGE:



POSTED INFORMATIONS

| Posted At | Area | View |
|---|---|---|
| 2020-08-02 | Area 1 | View |

## VIEW DETAILED INFORMATION PAGE:



INFORMATION

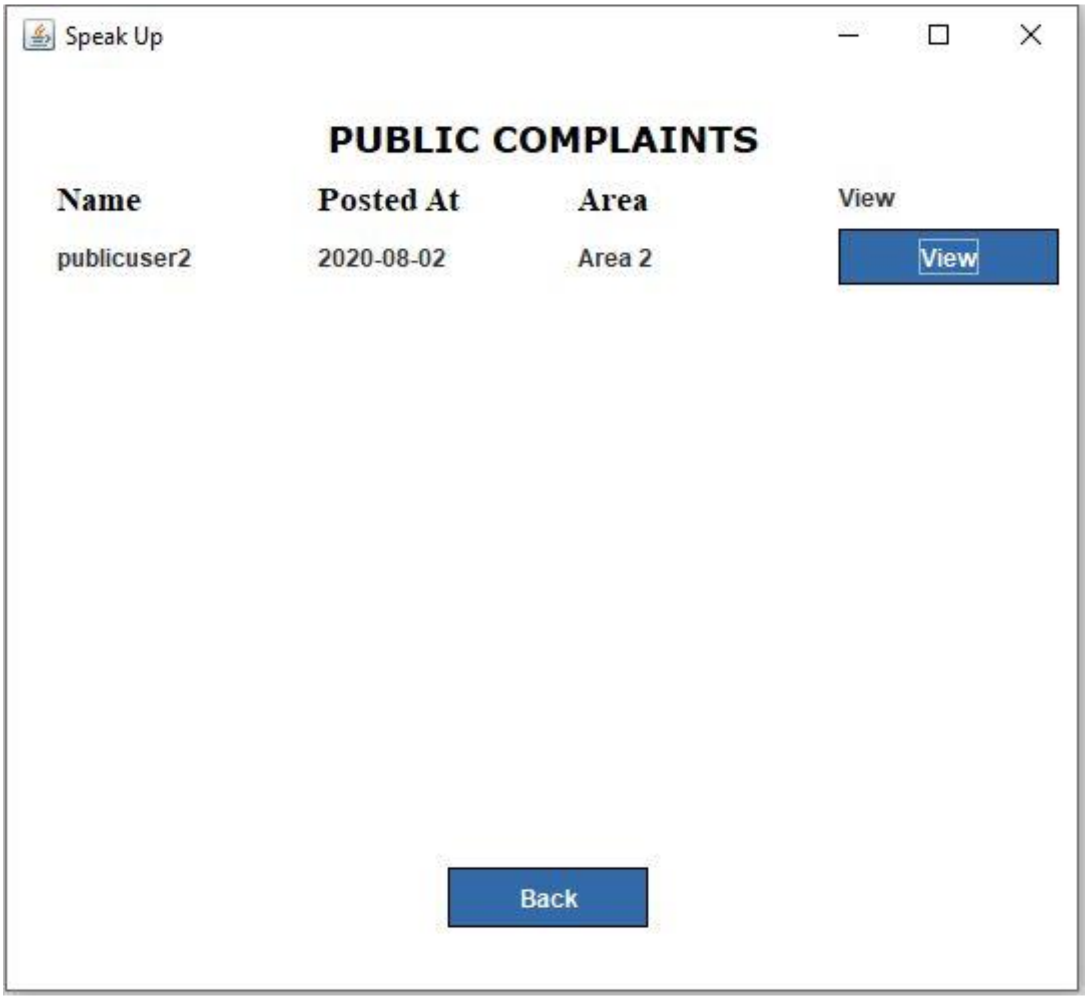
Sector: Corporation

City: City 1

Area: Area 1

Information: This is the Information.

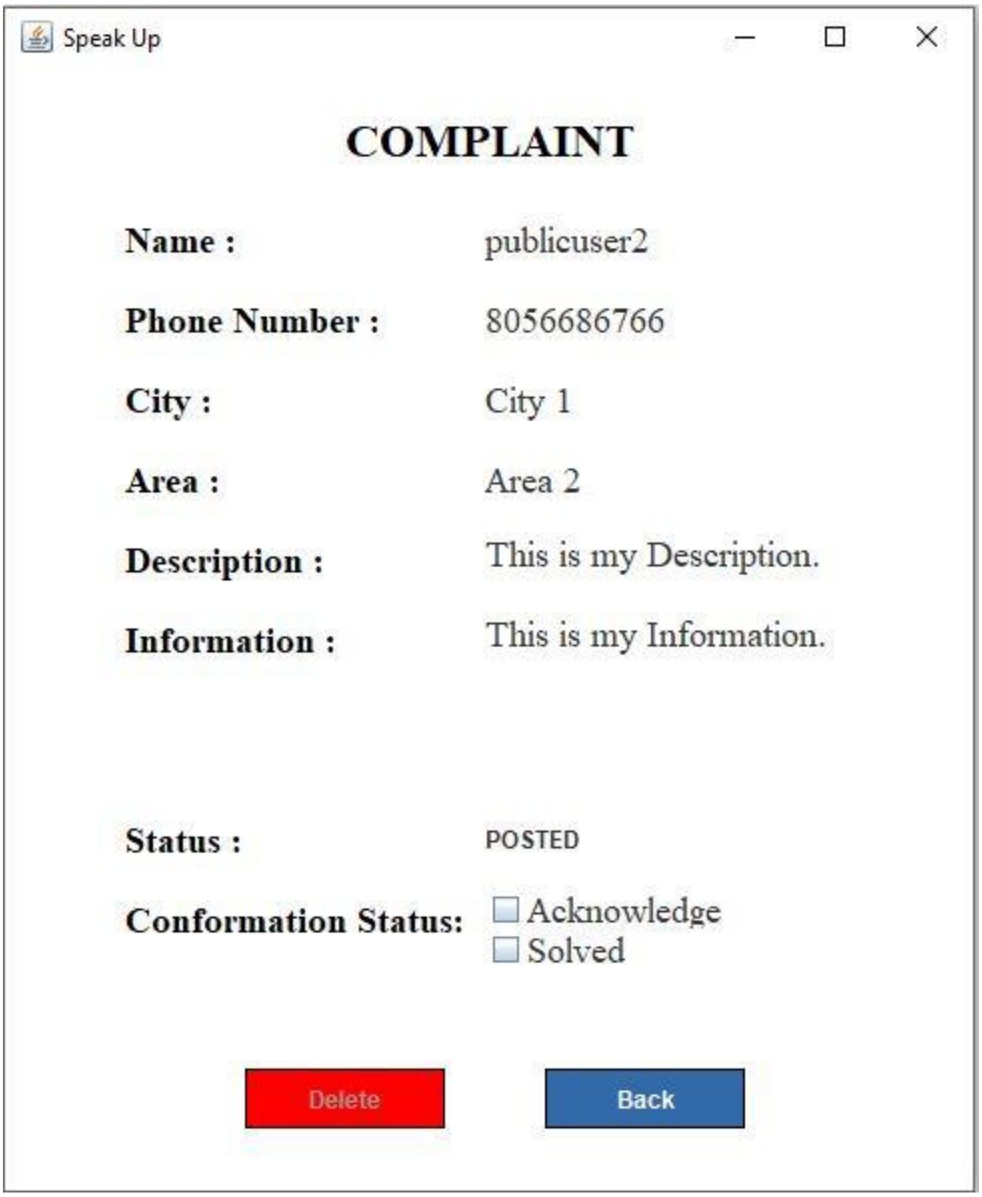
## VIEW COMPLAINTS PAGE:



PUBLIC COMPLAINTS

| Name | Posted At | Area | View |
|---|---|---|---|
| publicuser2 | 2020-08-02 | Area 2 | View |

## VIEW DETAILED COMPLAINT PAGE:

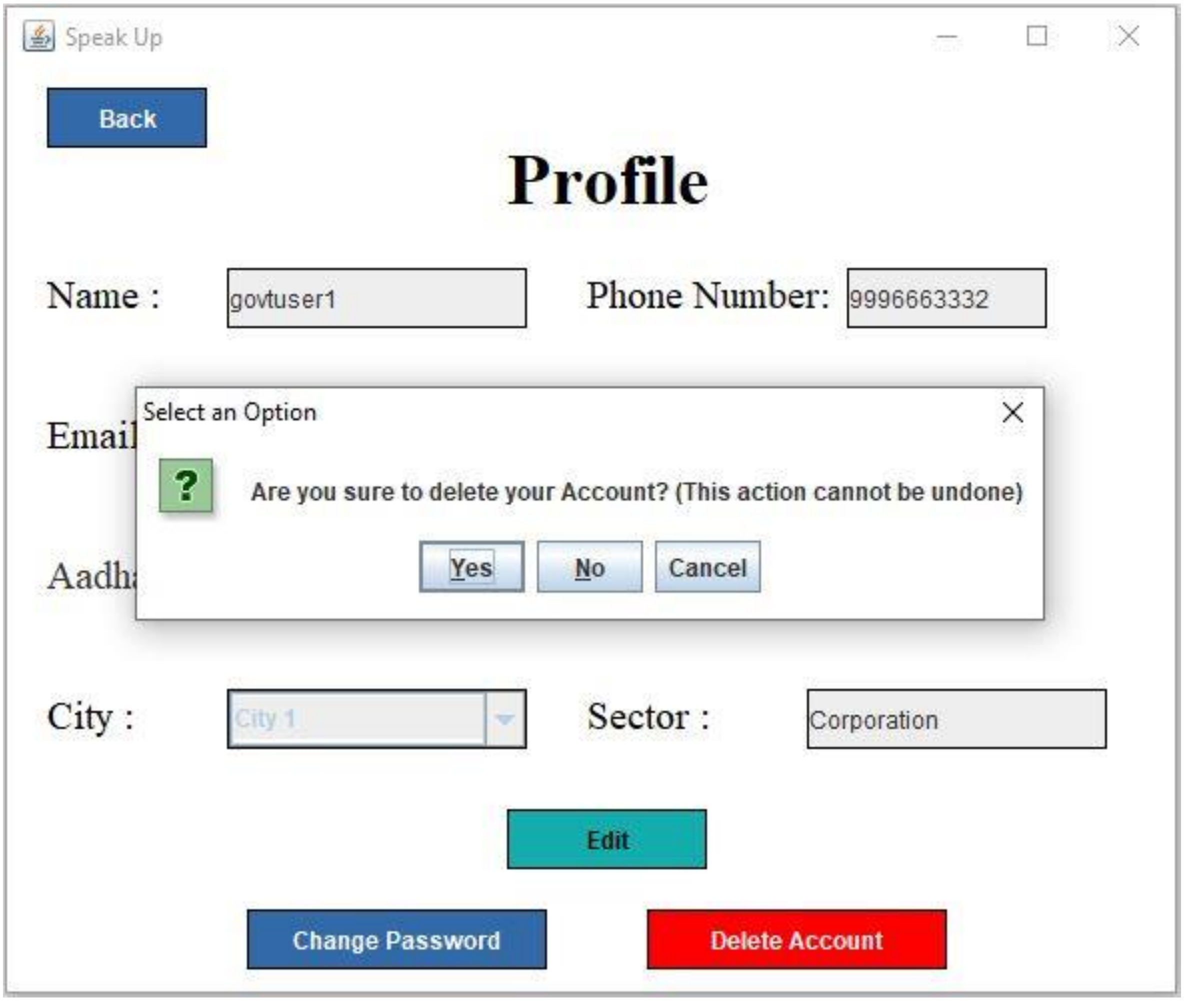


COMPLAINT

|  |  |
|---|---|
| **Name :** | publicuser2 |
| **Phone Number :** | 8056686766 |
| **City :** | City 1 |
| **Area :** | Area 2 |
| **Description :** | This is my Description. |
| **Information :** | This is my Information. |
| **Status :** | POSTED |
| **Conformation Status:** | □ Acknowledge<br>□ Solved |

## COMMON MODULE →

## CHANGE PASSWORD:



Profile

Name : govtuser1

Phone Number: 9996663332

Email

Aadh

Select an Option

Are you sure to delete your Account? (This action cannot be undone)

City : City 1

Sector : Corporation

**TEST CASES:**

**ADMIN LOGIN:**

| MODULES | TESTCASES | REMARKS | STATUS |
|---|---|---|---|
| Admin Login | Phone number=9965527556<br>Password=23@2000Raghu | Accepts and move to the Admin home page | Passed |
| Admin Login | Phone number=9487823296<br>Password=12@3456Admin | Accepts and move to the Admin home page | Passed |
| Admin Login | Phone number=9566576972<br>Password=12@3456Admin2 | Accepts and move to the Admin home page | Passed |
| Admin Login | Phone number=8248956222<br>Password=23@2000Raghu | Doesn't accept and notify us account with given phone number dose not exists. | Failed |
| Admin Login | Phone number=9487879452<br>Password=12@3456Admin | Doesn't accept and notify us account with given phone number dose not exists. | Failed |
| Admin Login | Phone number=8248591622<br>Password=12@3456Admin2 | Doesn't accept and notify us Invalid password. | Failed |
| Admin Login | Phone number=<EMPTY><br>Password=<EMPTY> | Doesn't accept and notify us This field cannot be empty. | Failed |

**ADMIN REGISTRATION:**

| MODULES | TESTCASES | REMARKS | STATUS |
|---|---|---|---|
| Admin Registration | Phone number=9965527556<br>Password=23@2000Raghu | Accept and notify us account created successfully | Passed |
| Admin Registration | Phone number=9487879452<br>Password=12@3456Admin | Accept and notify us account created successfully | Passed |
| Admin Registration | Phone number=9566895426<br>Password=12@3456Admin2 | Accept and notify us account created successfully. | Passed |
| Admin Registration | Phone number=2489562223<br>Password=23@2000Raghu | Doesn’t accept and notify us enter a valid phone number | Failed |
| Admin Registration | Phone number=9487879452<br>Password=12@3456admin | Doesn’t accept and notify us password must follow the given conditions. | Failed |
| Admin Registration | Phone number= 9566576972<br>Password=<EMPTY> | Doesn’t accept and notify us this field cannot be empty. | Failed |
| Admin Registration | Phone number= <EMPTY><br>Password=<EMPTY> | Doesn’t accept and notify us this field cannot be empty. | Failed |

## REGISTER GOVERNMENT OFFICIAL:

| MODULES | TESTCASES | REMARKS | STATUS |
|---|---|---|---|
| Government official Registration | Name=govtuser1<br>Phone number=9996663332<br>Email-Id=govtuser1@gov.in<br>Password=23@Govtuser1<br>Confirm password=23@Govtuser1<br>Aadhar number:12345789546<br>City=Select the city<br>Sector=Corporation | Accept and notify us account created successfully and redirect to login page. | Passed |
| Government official Registration | Name=govtuser1<br>Phone number=2996663332<br>Email-Id=govtuser1@gov.in<br>Password=23@Govtuser1<br>Confirm password=23@Govtuser1<br>Aadhar number:12345789546<br>City=Select the city<br>Sector=Corporation | Doesn't accept and notify us Enter a valid phone number. | Failed |
| Government official Registration | Name=govtuser1<br>Phone number=9996663332<br>Email-Id=govtuser1@gmail.com.<br>Password=23@Govtuser1<br>Confirm password=23@Govtuser1<br>Aadhar number:12345789546<br>City=Select the city<br>Sector=Corporation | Doesn't accept and notify us Enter a valid Email-Id. | Failed |
| Government official Registration | Name=govtuser1<br>Phone number=9996663332<br>Email-Id=govtuser1@gov.in<br>Password=23@govtuser1<br>Confirm password=23@govtuser1<br>Aadhar number:12345789546<br>City=Select the city<br>Sector=Corporation | Doesn't accept and notify us password must follow the given conditions. | Failed |

| | | | |
|---|---|---|---|
| Government official Registration | Name=govtuser1<br>Phone number=9996663332<br>Email-Id=govtuser1@gov.in<br>Password=23@Govtuser1<br>Confirm password=23@Govtuser1<br>Aadhar number:1234578956<br>City=Select the city<br>Sector=Corporation | Doesn’t accept and notify us Enter a valid aadhar number. | Failed |
| Government official Registration | Name=<EMPTY><br>Phone number=<EMPTY><br>Email-Id=<EMPTY><br>Password=<EMPTY><br>Confirm password=<EMPTY><br>Aadhar number: =<EMPTY><br>City=<EMPTY><br>Sector=<EMPTY> | Doesn’t accept and notify us This field cannot be empty. | Failed |

## GOVERNMENT OFFICIAL LOGIN:

| MODULES | TESTCASES | REMARKS | STATUS |
|---|---|---|---|
| Government Official Login | Phone number=9996663332<br>Password=23@Govtuser1 | Accepts and move to the Government official home page | Passed |
| Government Official Login | Phone number=9487823296<br>Password=12@3456govt | Doesn’t accept and notify us account with given phone number doesn’t exists. | Failed |
| Government Official Login | Phone number=9996663332<br>Password=12@3456govt | Doesn’t accept and notify us Invalid password. | Failed |
| Government Official Login | Phone number=<EMPTY><br>Password=<EMPTY> | Doesn’t accept and notify us This field cannot be empty. | Failed |

**REGISTER PUBLIC:**

| MODULES | TESTCASES | REMARKS | STATUS |
|---|---|---|---|
| Public Registration | Name=publicuser2<br>Email-Id=publicuser2@gmail.com<br>Gender=male<br>Password=23@Publicuser2<br>Confirm password=23@Publicuser2<br>Address=address2<br>Area=Select the area<br>City=Select the city<br>Pin code=638656<br>Phone number=8056686766 | Accept and notify us account created successfully and redirect to login page. | Passed |
| Public Registration | Name=publicuser2<br>Email-Id=publicuser2@gmail.com<br>Gender=male<br>Password=23@Publicuser2<br>Confirm password=23@Publicuser2<br>Address=address2<br>Area=Select the area<br>City=Select the city<br>Pin code=638656<br>Phone number=2056686766 | Doesn't accept and notify us Enter a valid phone number. | Failed |
| Public Registration | Name=publicuser2<br>Email-Id=publicuser2@gmail.com.<br>Gender=male<br>Password=23@Publicuser2<br>Confirm password=23@Publicuser2<br>Address=address2<br>Area=Select the area<br>City=Select the city<br>Pin code=638656<br>Phone number=8056686766 | Doesn't accept and notify us Enter a valid Email-Id. | Failed |

| | | | |
|---|---|---|---|
| Public Registration | Name=publicuser2<br>Email-Id=publicuser2@gmail.com<br>Gender=male<br>Password=23@publicuser2<br>Confirm password=23@publicuser2<br>Address=address2<br>Area=Select the area<br>City=Select the city<br>Pin code=638656<br>Phone number=8056686766 | Doesn’t accept and notify us password must follow the given conditions. | Failed |
| Public Registration | Name=publicuser2<br>Email-Id=publicuser2@gmail.com.<br>Gender=male<br>Password=23@Publicuser2<br>Confirm password=23@Publicuser2<br>Address=address2<br>Area=Select the area<br>City=Select the city<br>Pin code=6386562<br>Phone number=8056686766 | Doesn’t accept and notify us Enter a valid pin code. | Failed |
| Public Registration | Name=publicuser2<br>Email-Id=publicuser2@gmail.com.<br>Gender=male<br>Password=23@Publicuser2<br>Confirm password=23@publicuser2<br>Address=address2<br>Area=Select the area<br>City=Select the city<br>Pin code=6386562<br>Phone number=8056686766 | Doesn’t accept and notify us password must match. | Failed |

| | | | |
|---|---|---|---|
| Public Registration | Name=<EMPTY><br>Email-Id=<EMPTY><br>Gender=male<br>Password=<EMPTY><br>Confirm password=<EMPTY><br>Address=<EMPTY><br>Area=<EMPTY><br>City=<EMPTY><br>Pin code=<EMPTY><br>Phone number=<EMPTY> | Doesn’t accept and notify us this field cannot be empty. | Failed |

**PUBLIC LOGIN:**

| MODULES | TESTCASES | REMARKS | STATUS |
|---|---|---|---|
| Public Login | Phone number=8056686766<br>Password=23@Publicuser2 | Accepts and move to the Public home page | Passed |
| Public Login | Phone number=9487823296<br>Password=23@Publicuser2 | Doesn’t accept and notify us account with given phone number doesn’t exists. | Failed |
| Public Login | Phone number=8056686766<br>Password=23@publicuser2 | Doesn’t accept and notify us Invalid password. | Failed |
| Public Login | Phone number=<EMPTY><br>Password=<EMPTY> | Doesn’t accept and notify us This field cannot be empty. | Failed |

**PUBLIC MAKE COMPLAINT PAGE:**

| MODULES | TESTCASES | REMARKS | STATUS |
|---|---|---|---|
| Make Complaint | Sector=Select the Sector<br>Area=<Default area selected from Public model><br>Description=Description 1<br>Information=Information 1 | Accepts and notify us your complaint has been posted successfully and redirect to home page. | Passed |
| Make Complaint | Sector= Select the Sector<br>Area=<Default area selected from Public model><br>Description=<EMPTY><br>Information=Information 1 | Doesn't accept and notify us This field cannot be empty. | Failed |
| Make Complaint | Sector= Select the Sector<br>Area=<Default area selected from Public model><br>Description=Description 1<br>Information=<EMPTY> | Doesn't accept and notify us This field cannot be empty. | Failed |
| Make Complaint | Sector= Select the Sector<br>Area=<Default area selected from Public model><br>Description=<EMPTY><br>Information=<EMPTY> | Doesn't accept and notify us This field cannot be empty. | Failed |

## POST ANNOUNCEMENT PAGE:

| MODULES | TESTCASES | REMARKS | STATUS |
|---|---|---|---|
| Post Announcement | Sector=<Default sector selected from Public model><br>Area=Select the area<br>Information=Information 1 | Accepts and notify us your complaint has been posted successfully and redirect to home page. | Passed |
| Post Announcement | Sector=<Default sector selected from Public model><br>Area=Select the area<br>Information=<EMPTY> | Doesn't accept and notify us This field cannot be empty. | Failed |

## TARGET:

Due to this Pandemic situation (COVID-19), the Public feels unsafe to visit Government Offices to register their complaints in their area. To overcome this, we came up with an application that will be useful for both Public users and Government Officials. So, our TARGET is **PUBLIC** and **GOVERNMENT.**

It will be a simple application to register their complaints and it creates a **user-friendly** relationship between **Public users** and **Government Officials.**

## CONCLUSION:

This project will be of high value if deployed now because of the Covid-19 pandemic. The project is ready to be deployed and by using which any citizen of India could **Speak Up** for himself/herself easily with our project and his/her query shall be acknowledged and solved at once. We believe that this project is of much need now than ever.