

Case Study on

Multi-User Complaint Management System

Introduction

In today's fast-paced organizations and educational institutions, managing complaints efficiently is essential for maintaining transparency and trust.

CompSolveHub is a multi-user complaint management system designed to simplify the process of raising, tracking, and resolving issues through a centralized platform.

The application allows users to submit complaints and monitor their status in real time, while administrators can manage, update, and resolve issues effectively. With role-based access, email notifications, and an intuitive user interface

CompSolveHub ensures faster resolution, improved accountability, and a seamless communication flow between users and administrators.

Abstract

CompSolveHub is a web-based multi-user complaint management system developed to streamline the process of complaint registration, tracking, and resolution.

The system provides a centralized platform where users can raise complaints and monitor their status, while administrators efficiently manage and resolve issues through a dedicated dashboard.

Built using Spring Boot and a responsive user interface, the application ensures secure role-based access, real-time status updates, and automated email notifications. By improving transparency, accountability, and communication, CompSolveHub enhances organizational efficiency and delivers a structured approach to complaint handling.

What we are going to build: Client Requirement

- A **web-based complaint management system** accessible to multiple users.
- Facility for users to **raise complaints** with relevant details.
- **User registration and login system** with role-based access (Admin & User).
- Ability for users to **track complaint status** in real time.
- An **admin dashboard** to view, manage, and update complaints.
- **Status management** (Open, In Progress, Resolved) for complaints.
- **Automated email notifications** to users when complaint status changes.

What we are going to build: Some technical terms.

- A **Spring Boot**–based web application following the MVC architecture.
- RESTful controllers to handle user and admin operations.
- Role-based authentication and authorization (**Admin & User**).
- **MySQL** relational database for secure and structured data storage.
- **Bootstrap-powered** responsive UI for cross-device compatibility.
- Email service integration (**SMTP**) for automated status notifications.
- **CRUD** operations for complaint and user management.

What are the technologies and tools we are going use ?

- **Framework:** Spring Boot Java Framework
- **Front-end :** HTML5, CSS3, JS , Bootstrap
- **Programming Language:** Java 8+
- **Build Tool:** Maven
- **IDE:** Spring Tool Suite (STS)
- **Web Server:** Apache Tomcat
- **Spring Modules:** Spring Core, Spring Security (JWT), Spring Data JPA (Hibernate)
- **Database:** MySQL Database

System Requirements

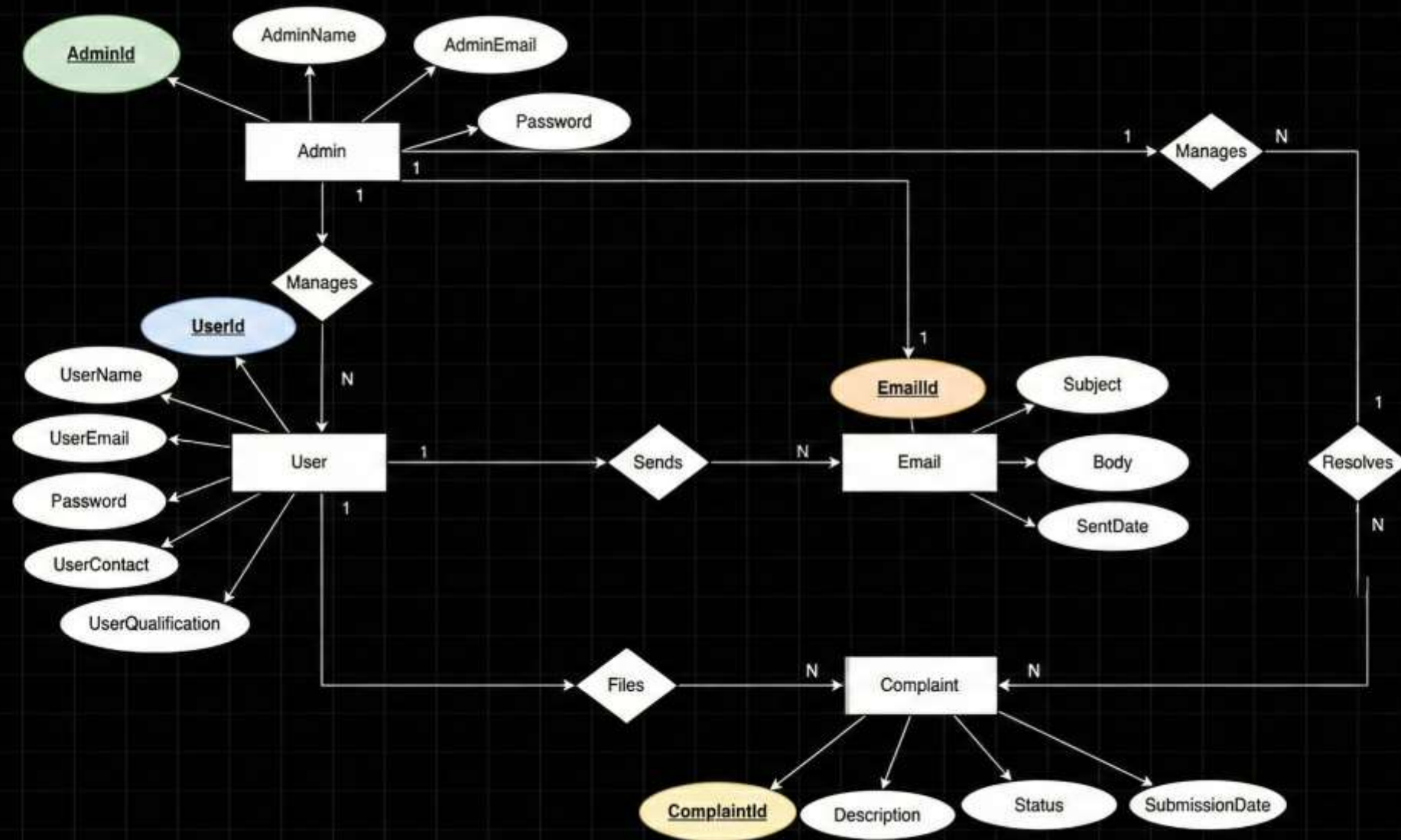
SOFTWARE REQUIREMENTS :

- **Back End:** Spring Boot, Spring Data JPA (Hibernate)
- **Database Server:** MySQL Server 8.0 CE
- **Web Server:** Apache Tomcat
- **IDE:** Spring Tool Suite (STS)
- **Browser:** Any web-browser
- **Testing Tool:** Junit(optional)

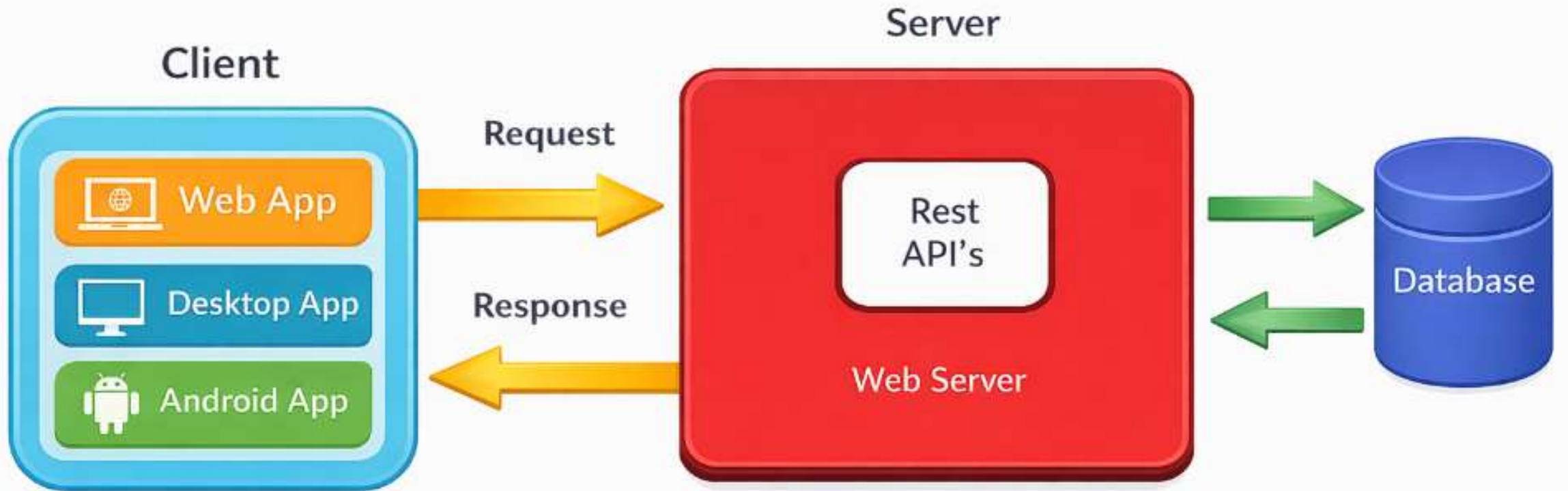
Project Module

- **Admin Module**
- **User Module**
- **Complaint Management Module**
- **Email Notification Module**

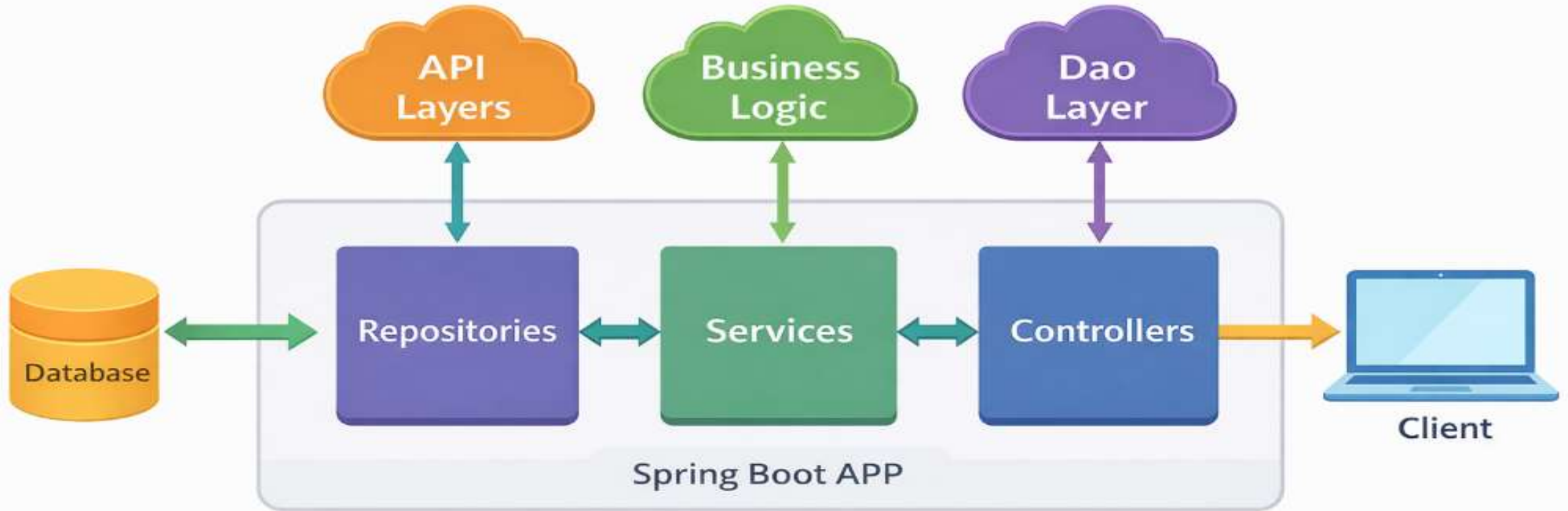
ER DIAGRAM



Client – Server Architecture



Architecture of Web App



ADMIN MODULE

In the Admin Module, the admin can access the system after successful login. The admin is responsible for managing the overall complaint handling process. Admin details are stored and managed through backend services.

- The Admin module allows the admin to perform **CRUD operations** using REST APIs. Admin details can be saved using **POST** Mapping, retrieved using **GET** Mapping, updated using **PUT** Mapping, and deleted using **DELETE** Mapping.
- The Admin module contains methods such as **Save Admin**, **View All Complaints**, **Update Complaint Status**, and **Delete Complaint** to ensure efficient management.
- The Admin has the authority to **view all user complaints**, update their status (Open, In Progress, Resolved), and monitor complaint resolution.
- After successful implementation of the Admin Module, child modules such as the User Module and Complaint Management Module are integrated under admin supervision.

USER MODULE

In the User Module, users can access the application after successful registration and login. This module allows users to interact with the system by submitting and tracking their complaints.

- The User module allows users to register and log in to the application securely. User details are stored and managed through backend services.
- Users can raise new **complaints** by submitting complaint details using **POST** Mapping. They can also view their submitted complaints using **GET** Mapping.
- Users can track the status of **complaints** (Open, In Progress, Resolved) in real time through the user dashboard.
- The User module contains methods such as **Register User**, **Login User**, **Create Complaint**, **View Complaints**, and **Track Complaint Status** to ensure smooth user interaction with the system.

Http Request Methods

HTTP defines a set of request methods to indicate the desired action to be performed for a given resource.

| | | |
|--------|---|--|
| GET | http://localhost:8080/admin/complaints | Returns the list of all complaints |
| GET | http://localhost:8080/admin/complaints/AC_issue | Returns the list of complaints raised by a specific user |
| POST | http://localhost:8080/user/create-complaint | Creates a new complaint |
| PUT | http://localhost:8080/admin/update-status/1 | Updates the status of a complain by complaint ID |
| DELETE | http://localhost:8080/user/delete-complaint/3 | Deletes a complaint by complaint ID |

COMPLAINT MANAGEMENT MODULE

- The Complaint Management Module is responsible for handling the complete lifecycle of complaints within the system.
- It allows users to **create** new complaints by submitting details such as title, description, and category using **POST Mapping**.
- The module enables users to **view** their submitted complaints along with current status using **GET Mapping**.
- Admin users can **view all complaints** raised by different users through the admin dashboard.
- Admins can **update** the complaint status (Open, In Progress, Resolved) using **PUT Mapping** to track resolution progress.

EMAIL NOTIFICATION MODULE

- The Email Notification Module is used to provide **automated email communication** between the system and users.
- This module sends an **email notification to users** whenever the status of their complaint is updated by the admin.
- Email alerts are triggered when a complaint is resolved or moved to a different **status** such as Open or In Progress.
- The module is implemented using **SMTP (JavaMailSender)** for sending emails securely.

Tables for the database

The screenshot shows a database management interface with a left sidebar and a main SQL editor area.

Left Sidebar:

- Navigator:**
 - SCHEMAS:** Filter objects. A tree view shows 'college' and 'complaint_db'. Under 'complaint_db', 'Tables' is expanded, showing 'complaints' and 'users'. Other categories like Views, Stored Procedures, and Functions are also listed.
 - Administration** and **Schemas** tabs are at the bottom.
- Information:** Displays 'Schema: complaint_db'.

Main SQL Editor:

- Tab: **SQL File 3***
- Toolbar: Includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown.
- SQL Code:

```
1 • drop database complaint_db;
2
3 • CREATE DATABASE complaint_db;
4 • USE complaint_db;
5
6 • show tables;
7
8
9
10
11
```

Bottom Panel:

- Buttons: **Result Grid**, **Filter Rows:** (with a text input), **Export:** (with a printer icon), **Wrap Cell Content:** (with a text icon).
- Table List:**
 - Header: **Tables_in_complaint_db**
 - Row 1: **complaints** (with a collapse icon)
 - Row 2: **users** (highlighted)

Admin Database

Navigator

SCHEMAS

Filter objects

college

complaint_db

Tables

complaints

users

Views

Stored Procedures

Functions

crud

dummy

eshop

Administration Schemas

Information

Schema: complaint_db

SQL File 3

Limit to 1000 rows

1 • drop database complaint_db;

2

3 • CREATE DATABASE complaint_db;

4 • USE complaint_db;

5

6 • select * from users where role= "ADMIN";

7

8

9

10

11

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

| | id | email | name | password | role |
|---|------|-----------------|-------|----------|-------|
| ▶ | 2 | admin@gmail.com | Admin | admin123 | ADMIN |
| * | NULL | NULL | NULL | NULL | NULL |

Student Database

Navigator

SCHEMAS

Filter objects

college

complaint_db

Tables

complaints

users

Views

Stored Procedures

Functions

crud

dummy

eshop

Administration Schemas

Information

Schema: complaint_db

SQL File 3*

Limit to 1000 rows

1 • drop database complaint_db;

2

3 • CREATE DATABASE complaint_db;

4 • USE complaint_db;

5

6

7 • select * from users where role= "USER";

8

9

10

11

12

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

| | id | email | name | password | role |
|---|------|----------------------|--------|-----------|------|
| ▶ | 1 | hari@gmail.com | Hari | 12345 | USER |
| | 3 | kesh@gmail.com | kesh | 123456 | USER |
| | 4 | hariyt0910@gmail.com | AI Man | Hari1011@ | USER |
| ● | NULL | NULL | NULL | NULL | NULL |

Complaint Database

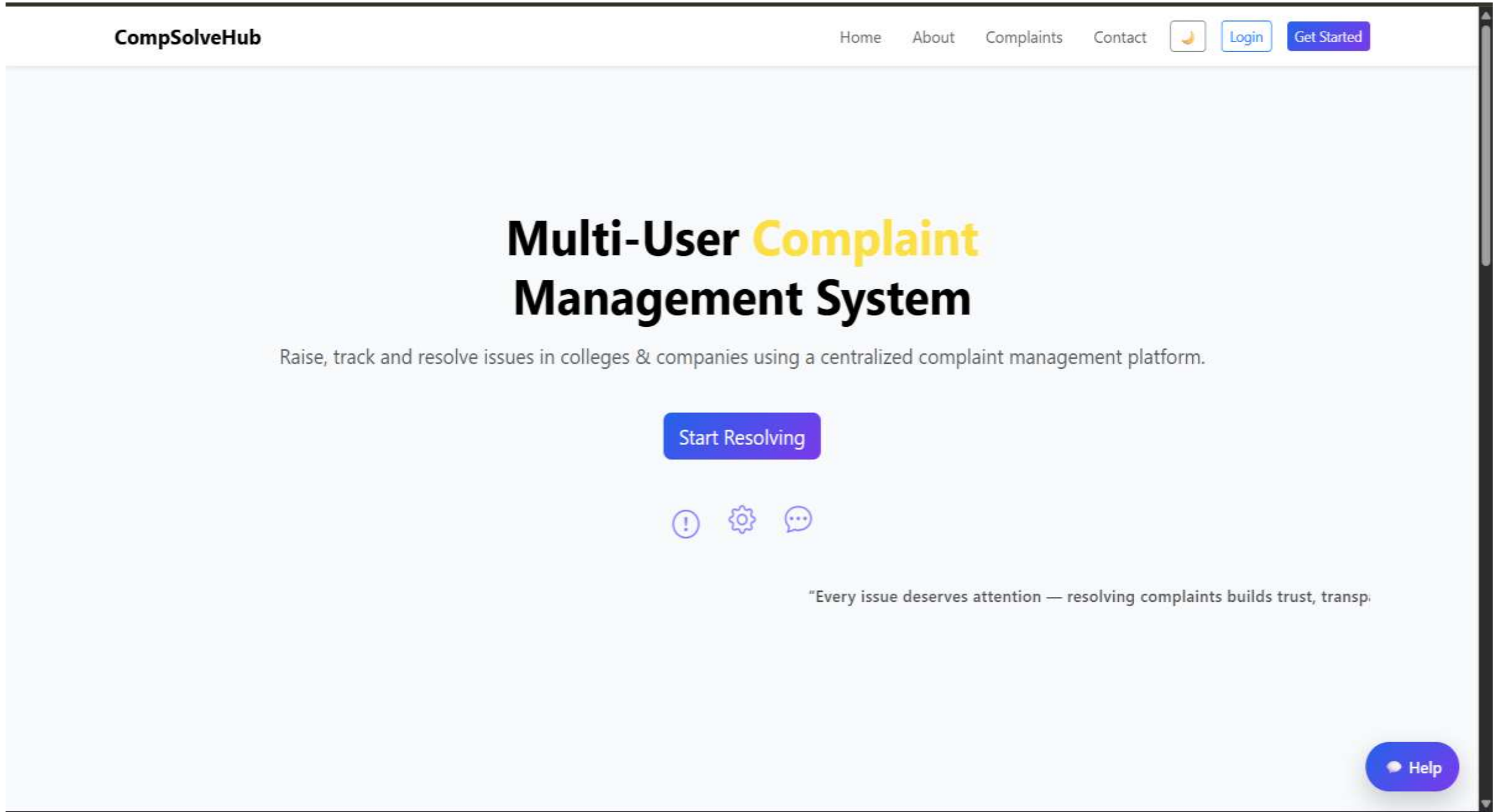
The screenshot displays a database management interface. On the left, the 'Navigator' pane shows a tree view of schemas, with 'complaint_db' selected. Below it, the 'Administration' and 'Schemas' tabs are visible, and the 'Information' pane shows 'Schema: complaint_db'. The main area is titled 'SQL File 3' and contains the following SQL queries:

```
1 • drop database complaint_db;
2
3 • CREATE DATABASE complaint_db;
4 • USE complaint_db;
5
6
7 • select * from complaints;
8
9
10
11
12
13
14
15
16
```

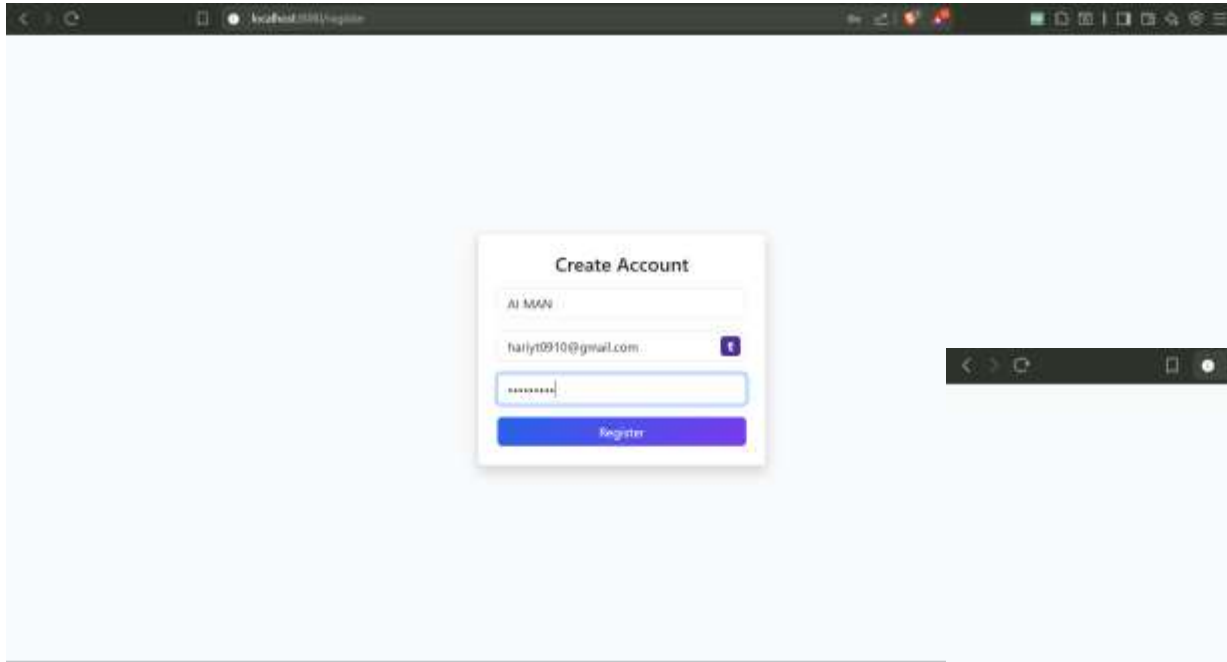
Below the SQL editor, the 'Result Grid' shows the output of the query. The grid has columns for 'id', 'created_at', 'description', 'resolved_at', 'status', 'title', and 'user_id'. The data is as follows:

| | id | created_at | description | resolved_at | status | title | user_id |
|---|----|----------------------------|---|----------------------------|----------|-------------------------|---------|
| ▶ | 1 | 2025-12-13 23:47:23.865736 | Boys issue in 301 | 2025-12-15 11:40:05.100230 | RESOLVED | harassment | 1 |
| | 2 | 2025-12-14 11:54:42.230509 | Please resolve the issue it is so hard to sit | 2025-12-14 11:56:16.990137 | RESOLVED | No proper working of AC | 1 |
| | 3 | 2025-12-14 13:20:10.338672 | not working | 2025-12-14 13:20:27.940537 | RESOLVED | fan | 1 |
| | 4 | 2025-12-14 14:46:00.575785 | Useless fights for girls | 2025-12-14 15:02:51.233645 | RESOLVED | fighting | 1 |
| | 5 | 2025-12-14 15:02:22.373574 | so hot and no aeration | 2025-12-15 11:10:35.623500 | RESOLVED | fan not working | 1 |
| | 6 | 2025-12-14 16:11:37.126897 | flickering | 2025-12-14 16:12:41.524374 | RESOLVED | lights | 1 |
| | 7 | 2025-12-15 11:42:11.253219 | The ethernet port is not working | 2025-12-15 11:45:49.554069 | RESOLVED | PC issue | 4 |
| | 8 | 2025-12-15 12:04:16.605162 | broken glasses, need to be replaced and fixed | NULL | OPEN | window issue | 1 |

ScreenShots of the Capstone Project



Create Account Page and Login Page



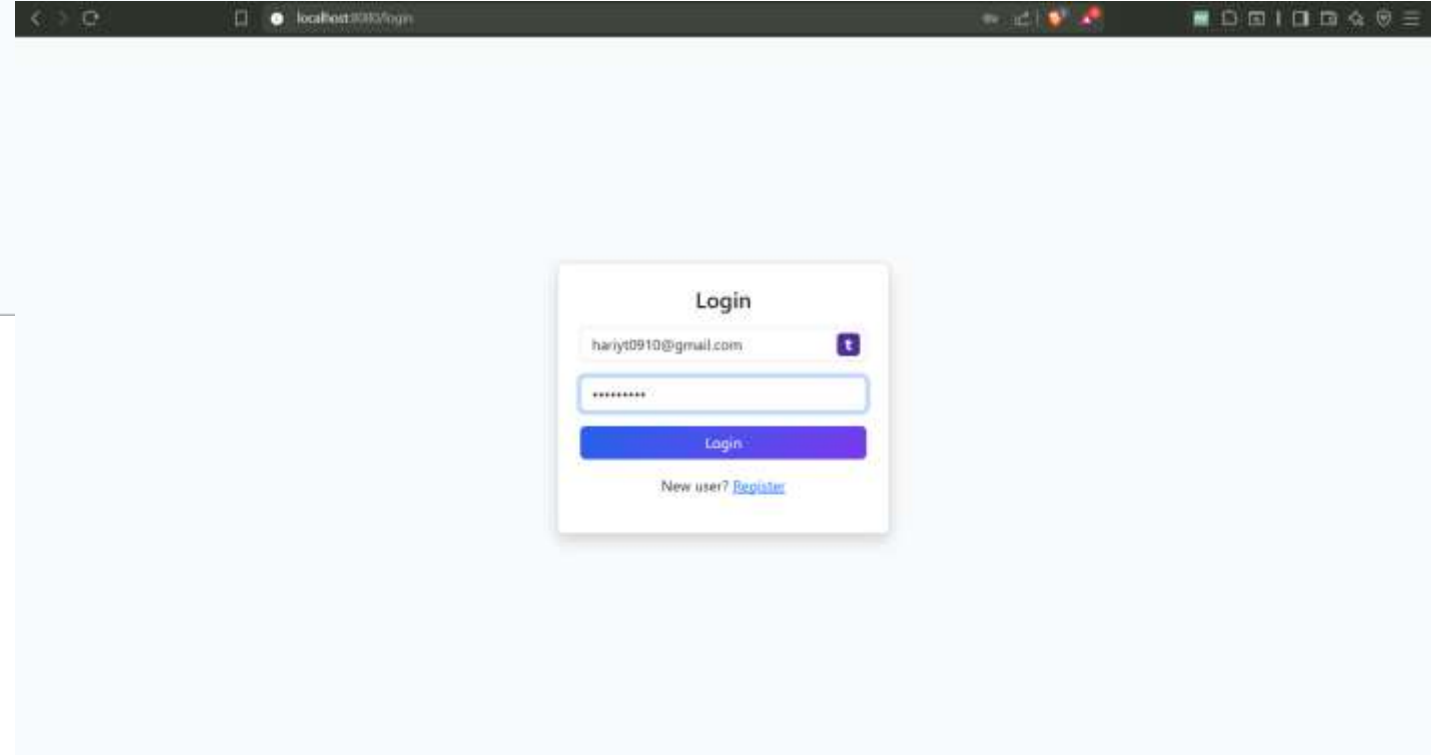
localhost:3030/register

Create Account

AI KAAA

hariyt0910@gmail.com

Register



localhost:3030/login

Login

hariyt0910@gmail.com

Login

New user? [Register](#)

Admin Dashboard

User Dashboard

Logout

+ Raise Complaint

| Title | Status | Created |
|-------------------------|----------|----------------------------|
| harassment | RESOLVED | 2025-12-13T23:47:23.865736 |
| No proper working of AC | RESOLVED | 2025-12-14T11:54:42.230509 |
| fan | RESOLVED | 2025-12-14T13:20:10.338672 |
| fighting | RESOLVED | 2025-12-14T14:46:00.575785 |
| fan not working | RESOLVED | 2025-12-14T15:02:22.373574 |
| lights | RESOLVED | 2025-12-14T16:11:37.126897 |
| window issue | OPEN | 2025-12-15T12:04:16.605162 |
| ac issue | RESOLVED | 2025-12-15T16:16:56.227973 |
| ac issues | OPEN | 2025-12-16T09:56:23.208158 |

User Dashboard

Admin Panel

Logout

Total Complaints

11

Open

2

In Progress

0

Resolved

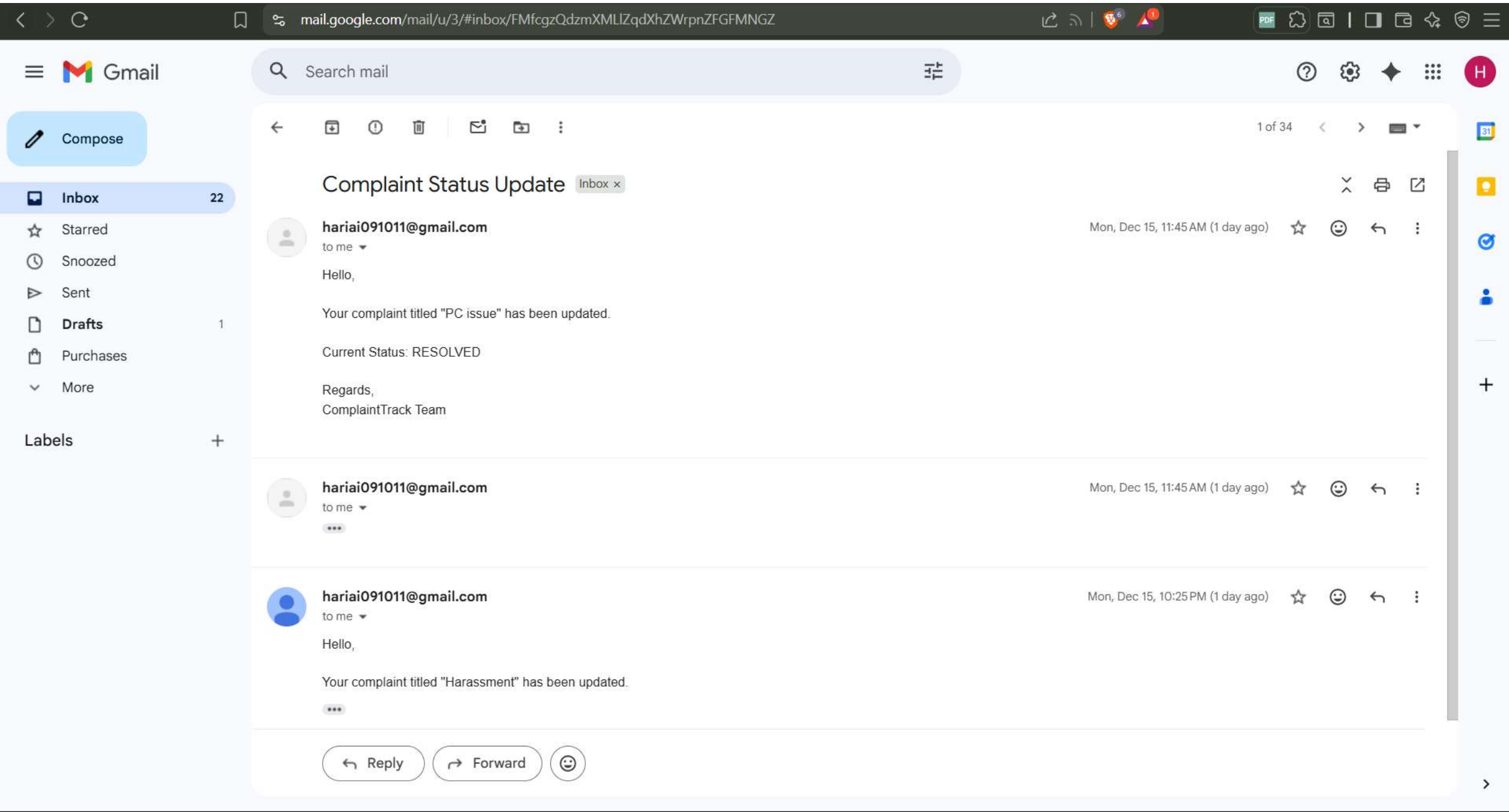
9

Search by complaint or user name

All Status

| Complaint | User | Status | Update |
|-------------------------|------|----------|--|
| harassment | Hari | RESOLVED | <div>RESOLVED</div> <div>Update Status</div> |
| No proper working of AC | Hari | RESOLVED | <div>OPEN</div> <div>Update Status</div> |
| fan | Hari | RESOLVED | <div>OPEN</div> <div>Update Status</div> |
| fighting | Hari | RESOLVED | <div>OPEN</div> <div>Update Status</div> |
| fan not working | Hari | RESOLVED | <div>OPEN</div> <div>Update Status</div> |
| lights | Hari | RESOLVED | <div>OPEN</div> <div></div> |

Email Status



GOOGLE DRIVE LINK OF DEMO VIDEO

<https://drive.google.com/drive/folders/1Wt70r9VV-8YDuc9eatOABoHjjyvSBWcx?usp=sharing>