**ONLINE COMPLAINT PORTAL**

**A PROJECT REPORT**

***Submitted by:***

**Team No. : 18**

**Sujal Wainganakar (221B400)**

**Tulja Rathod (221B413)**

**Utsav Rai (221B425)**

***Submitted To:***

**Dr. Amit Kumar Srivastava**



May-2025

***Software Engineering Lab***

**Department of Computer Science & Engineering**

**JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY, AB**

**ROAD, RAGHOGARH, DT. G A-473226 MP, INDIA**

# JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY

**Grade 'A+'** Accredited with by NAAC & Approved U/S 2(f) of the UGC Act,1956

A.B. Road, Raghogarh, Dist: Guna (M.P.) India, Pin-473226

Phone:07544 267051, 267310-14, Fax: 07544 267011

Website: [www.juet.ac.in](http://www.juet.ac.in/)

**DECLARATION BY THE STUDENTS**

We hereby declare that the work reported in the B. Tech. the project entitled **"Online Complaint Portal", "** as our 6th semester Software Engineering project for degreeof B.Tech (CSE) in Jaypee University of Engineering & Technology, Guna. As per best of our knowledge and belief there is no infringement of the intellectual property right and copyright. In case of any violation, we will solely be responsible.

**Sujal Wainganakar (221B400)**

**Tulja Rathod (221B413)**

**Utsav Rai (221B425)**

**Date: 13/05/2025**

Department of Computer Science and Engineering

Jaypee University of Engineering and Technology,

Guna

# JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY

**Grade 'A+'** Accredited with by NAAC & Approved U/S 2(f) of the UGC Act,1956

A.B. Road, Raghogarh, Dist: Guna (M.P.) India, Pin-473226

Phone:07544 267051, 267310-14, Fax: 07544 267011

Website: [www.juet.ac.in](http://www.juet.ac.in/)

This is to certify that the work titled **“Online Complaint Portal”** is submitted by **"Sujal Wainganakar, Tulja Rathod & Utsav Rai"** as their 6th semester Software Engineering project for degreeof B.Tech (CSE) of Jaypee University of Engineering & Technology, Guna. As per best of our knowledge and belief, there is no infringement of the intellectual property rights and copyright. Also, this work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

In case of any violation concern, students will solely be responsible.

**Signature of Supervisor**

**Dr. Amit Kumar Srivastava**

**Assistant Professor,**

**Department of CSE**

**Date: 13/05/2025**

## ACKNOWLEDGEMENT

We extend our heartfelt gratitude to everyone who facilitated the successful completion of this project. Our sincere thanks go to Dr. Amit Shrivastav Sir, whose invaluable guidance, insightful suggestions, and unwavering encouragement supported us throughout the development process and in compiling this report.

We are deeply appreciative of the time and effort you dedicated to reviewing and rectifying our numerous errors, contributing significantly to the refinement of our work.

Furthermore, we express our profound gratitude to our parents and friends for their unwavering support and assistance, which played a crucial role in the timely completion of this web project.

Lastly, we extend our thanks to all the team members of "**Online Complaint Portal**" for their collaboration and contributions, which were instrumental in the success of this endeavor.

**Sujal Wainganakar (221B400)**

**Tulja Rathod (221B413)**

**Utsav Rai (221B425)**

**Date : 13-05-2025**

### EXECUTIVE SUMMARY

The **Online Complaint Portal** is a web-based software system designed to streamline the process of lodging and managing public grievances. Developed as part of a Software Engineering project, this portal aims to bridge the gap between citizens and government officers by providing a transparent and efficient communication channel.

Through the portal, **citizens can register, log in, file complaints**, and track the resolution status. Simultaneously, **government officers can register, view complaints assigned to them**, update complaint statuses, and close cases upon resolution. A structured feedback mechanism also allows for further improvement based on user suggestions.

The system follows the **Incremental Process Model**, allowing the core features to be developed first, with additional functionalities added based on user feedback. Major components of the system include user interfaces for citizens and officers, a complaint management dashboard, and a relational database that maintains all user and complaint records.

The technology stack used includes **ReactJs, Tailwind CSS, NodeJs**, and **MySQL**. Risk management strategies, function point estimations, and ER diagrams were also included in the development documentation to ensure a thorough engineering process.

Overall, the portal enhances accessibility, accountability, and efficiency in grievance redressal systems, with room for future upgrades like multilingual support, mobile access, and notification features.

| **Table of Contents** |
| --- |
| **TITLE PAGE NO.** |
| Declaration of the student |
| Certificate of the Guide |
| Acknowledgment |
| Executive Summary |
| List of Figure |
| **Chapter-1 INTRODUCTION**  8 |
| 1.1 Aim & Objective |
| 1.2 Scope of the Project |
| 1.3 Problem Definition |
| 1.4 Project Overview |
| **Chapter-2 System Requirements Specification** 12 |
| 2.1 Introduction |
| 2.2 Overall Description |
| 2.3 Functional Requirements |
| 2.4 Non -Functional Requirements |
| 2.5 Other Requirements |
| **Chapter-3 System Analysis and Design** 15 |
| 3.1 Introduction |
| 3.2 Software Development Model |
| 3.3 Modules |
| 3.4 Analysis Model |
| 3.5 UML Diagrams |
| **Chapter-4 Technology Description & Architecture** 24 |
| 4.1 Architecture |
| 4.2 Component Diagram Overview |
| 4.3 Database |
| **Chapter-5 RESULTS/OUTPUTS** 29 |
| 5.1 Functional Results |
| 5.2 Sample Input-Output Observations |
| 5.3 Outcome Screens |
| 5.4 Performance Metrics |
| 5.5 User Feedback Summary |
| 5.6 Project Interfaces |
| **Chapter-6 CONCLUSIONS 37** |
| **Chapter-**7 **REFERENCES 38** |

**CHAPTER- 1**

**INTRODUCTION**

**1.1 Aim & Objectives**

The *Online Complaint Portal* project aims to provide a seamless and digital solution for lodging and managing public complaints, particularly for civic or administrative issues. This system enables citizens to file grievances online and allows government officials to address them efficiently through a centralized portal. The objective is to reduce manual processes, improve transparency, and enhance communication between the public and authorities.

Key objectives include:

* Allowing citizens to register and file complaints through a user-friendly web interface.
* Enabling government officers to receive, manage, and respond to complaints digitally.
* Tracking complaint status in real-time and notifying users of updates.
* Maintaining a structured database of users, complaints, and resolutions.
* Offering feedback mechanisms for users and officials to suggest improvements.
* Promoting accountability and quicker grievance redressal through digital means.

This project strives to bridge the gap between citizens and governance by streamlining the complaint resolution process in an efficient, transparent, and scalable manner.

**1.2 Scope of the Project**

The *Online Complaint Portal* covers the complete lifecycle of a complaint, from submission by the citizen to resolution and confirmation. It is intended to be a web-based system, accessible on desktop and mobile devices, with distinct modules for both citizens and government officers.

The scope includes:

**User Registration & Authentication:**

* Account creation and login functionality for both citizens and officers.
* Secure password-protected access.

**Complaint Management:**

* Interface for users to submit complaints, including photo and description.
* Auto-assignment of complaints to relevant officers based on area.

**Complaint Tracking:**

* Real-time status updates (Pending, Resolved, Closed).
* Citizen confirmation for closure of complaints.

**Administrative Functions:**

* Officer dashboard to view assigned complaints and update status.
* Feedback tools for users and staff to improve the platform.

**Platform Support & Scalability:**

* Web-based access with potential for future mobile app integration.
* Designed to support large user volumes and scalable for other regions.

**1.3 Problem Definition**

Despite digital advancements, the traditional complaint lodging process in many public departments is often manual, slow, and lacks transparency. Citizens frequently face delays, unresponsiveness, and poor follow-up, while officials struggle with paper-based systems and disorganized workflows.

The *Online Complaint Portal* addresses the following issues:

* **Inefficient Complaint Handling**: Manual processes lead to delays and data loss.
* **Lack of Transparency**: Users have limited visibility into the status of their complaints.
* **Limited Accessibility**: People must visit government offices physically to report issues.
* **No Feedback Mechanism**: There is often no structured way for users to give feedback or rate services.
* **Workload Imbalance**: Complaints are not fairly distributed among officers.

By introducing a digital portal, the project ensures fast, reliable, and trackable complaint resolution, thereby fostering better governance and citizen satisfaction.

**1.4 Project Overview**

The *Online Complaint Portal* is a web-based application developed using standard web technologies. It integrates user interaction, complaint management, database handling, and administrative tools to provide a comprehensive complaint redressal solution.

**System Features:**

* **Frontend**: ReactJs and Tailwind CSS
* **Backend**:Node.js-based logic for handling user input, processing complaints and updating.
* **Database**: MySQL for storing user profiles, complaint data, and resolution statuses.

**Core Functional Modules:**

* **User Side**:
  + Register/Login.
  + File complaint with area, photo, and description.
  + View and track complaint status.
  + Confirm resolution and provide feedback.
* **Officer Side**:
  + View complaints assigned by area.
  + Update status to “Resolved” or “Closed.”
  + View statistics of complaints handled.
* **System Logic**:
  + Auto-assignment based on area and officer workload.
  + Feedback-based improvements and future analytics.

**Use Cases:**

* **Urban Municipalities**: Report broken roads, garbage issues, etc.
* **Educational Institutes**: Students can file maintenance complaints.
* **Public Offices**: Citizens raise service-related grievances digitally.

**Block Diagram**





Assign Officer





Solve the problem





Check Status





Customer Feedback



**CHAPTER-2**

**SYSTEM REQUIREMENTS SPECIFICATION**

**1. Introduction**

**1.1 Purpose**

The Online Complaint Portal (OCP) is a web-based application designed to allow citizens to file complaints and report issues to the appropriate government authorities. The system streamlines complaint management by enabling officers to track, update, and resolve complaints efficiently.

**Goals:**

* Reduce manual effort and paperwork in complaint handling.
* Provide 24/7 access to the complaint portal for seamless issue reporting.
* Ensure accurate tracking of submitted, in-progress, and resolved complaints.

**1.2 Document Conventions**

* **UI Screens:** Wireframe diagrams.
* **Dates:** yyyy-mm-dd format.
* **Error Codes:** Use HTTP 4xx for client errors and HTTP 5xx for server errors.

**1.3 Intended Audience**

* **Citizens**: To file complaints and track their resolution status.
* **Government Officers**: To receive, process, and resolve complaints.
* **Developers**: To implement and maintain the system.

**1.4 Project Scope**

The system will:

* Enable users to register and log in.
* Allow citizens to submit complaints with necessary details.
* Assign complaints to relevant government officers based on location.
* Enable officers to update and resolve complaints.
* Notify users about complaint progress.
* Maintain a database of complaints and resolutions.

**1.5 References**

* IEEE 830-1998 SRS Standard.
* ISO/IEC 27001 for data security.
* University library policies.

**2. Overall Description**

**2.1 Product Perspective**

Ensure accurate tracking of submitted, in-progress, and resolved complaints.

**System Architecture:**

* **Frontend:** React.js (Web-based)
* **Backend:** Node.js with REST APIs
* **Database:** MySQL

**2.2 Product Functions**

* **User Management:** Registration and login for citizens and officers.
* **Complaint Submission**: Citizens can submit complaints with descriptions and media attachments.
* **Complaint Assignment**: The system automatically assigns complaints to officers based on location.
* **Complaint Tracking**: Citizens can track the status of their complaints.
* **Status Updates**: Officers can update the status of complaints (Pending, Resolved, Closed).
* **Notifications**: Citizens receive notifications on status changes.
* **Feedback Mechanism**: Citizens can provide feedback on complaint resolutions.

**2.3 User Characteristics**

* **Citizens**: Basic knowledge of using web applications.
* **Government Officers**: Intermediate technical skills to handle complaints and updates.

**2.4 Constraints**

* System only works on modern web browsers.
* Complaints are limited to specific geographical areas.
* No multilingual support.

**2.5 Assumptions and Dependencies**

* Users must have valid identification details..
* Regular database backups will be maintained.

.

**3. Specific Requirements**

**3.1 Functional Requirements**

| **ID** | **Requirement Description** | **Priority** |
| --- | --- | --- |
| FR-1 | Users can register and log in using their credentials. | High |
|  |  |  |
| FR-2 | Citizens can file complaints with images/videos. | High |
| FR-3 | Officers can view and update complaints. | High |
| FR-4 | The system assigns complaints based on location. | Medium |
| FR-5 | Citizens can track complaint progress. | High |
| FR-6 | Notifications sent on status updates. | Medium |
| FR-7 | Users can provide feedback on complaint resolution. | Low |

**3.2 External Interface Requirements**

**3.2.1 User Interfaces:**

* **Citizen Dashboard**: File complaints, track status.
* **Officer Dashboard**: Manage assigned complaints.

**3.2.2 Hardware Interfaces:**

* Server: Minimum 8 GB RAM, 500 GB storage.
* User Devices: Any device with a modern web browser.

**3.2.3 Software Interfaces:**

* Web-based authentication system.
* MySQL for data storage.

**3.3 System Features**

**Feature 1: User Authentication**

* **Input**: User credentials (username, password).
* **Output**: Access to dashboard on successful login.

**Feature 2: Complaint Submission**

* **Input**: Title, description, location(pin-code), attachments.
* **Output**: Complaint ID assigned, notification sent.

**Feature 3: Status Update**

* **Input**: Complaint ID, status change.
* **Output**: Citizen notified of progress.

**3.4 Non-Functional Requirements**

| **Category** | **Requirement** |
| --- | --- |
| Performance | Complaint submission must process within 3 sec. |
| Security | Encrypt user data using AES-256. |
| Availability | System uptime of 99.9% annually. |
| Usability | Mobile-friendly and accessible UI. |
| Maintainability | Modular architecture for future expansion. |

**3.5 Other Requirements**

* Compliance with data privacy laws.
* Daily system backups.nces assignments to prevent overloading specific officers

**Chapter-3**

**System Analysis and Design**

**3.1 Introduction**

System analysis is a crucial phase in the development of any software project, as it lays the foundation for understanding the problem domain, identifying user needs, and defining clear system requirements. For the development of the Employee Management System, the Iterative Waterfall Model was chosen as the software development methodology. This model follows a phased approach similar to the traditional waterfall model, but with the flexibility to revisit and refine previous stages based on feedback or new insights. Each phase including requirement gathering, system design, implementation, testing, and deployment is completed in sequence, but with iterations allowed where needed for improvement. The Iterative Waterfall Model is suitable for this project because the system’s core requirements are largely well-defined, yet the model allows adaptability if minor adjustments are required during development. Its structured and feedback-oriented nature ensures thorough documentation, which is particularly beneficial for academic projects, long-term maintenance, and future scalability. During the analysis phase, detailed requirements were collected and organized into functional and non-functional categories. These requirements were then modeled using diagrams such as use case diagrams, activity diagrams, and data flow diagrams to provide a visual understanding of system behavior. Additionally, the core system modules were identified and broken down for easier management in the design and development stages. This structured analysis ensures that the final system meets user expectations and performs all intended operations efficiently, reducing the risk of costly changes in later stages.

**3.2 Software Development Model**

For the development of the Employee Management System, the Iterative Waterfall Model has been selected as the approach. The Iterative Waterfall Model follows a structured, phase-by phase development cycle like the traditional waterfall model but introduces the flexibility to revisit and refine earlier stages based on feedback or issues discovered in later phases. This method is ideal for projects with mostly well-defined requirements but where incremental improvements may be needed during development. It ensures that each stage of development—whether it's planning, design, or implementation—is well-documented and systematically revisited if necessary. 17 The decision to use the Iterative Waterfall Model stems from its balance of structure and adaptability, making it especially suitable for academic and moderately complex projects like this one. By following a logical progression from one phase to the next while allowing revisions, we ensure that every requirement is fully understood and addressed, and any emerging gaps can be corrected early—minimizing the risk of major errors or rework at later stages Iterative Waterfall Model:

- **Requirement Analysis**: In this phase, all functional and non-functional requirements of the system are gathered and documented. Stakeholder needs and system expectations are clearly defined.

**- System Design**: Based on the gathered requirements, the system’s architecture is designed. This includes designing the database structure, user interface layout, and system modules.

- **Implementation**: The system is developed according to the design. Coding is done using selected technologies (Django for backend, HTML for frontend).

- **Testing**: The system is thoroughly tested to identify and fix any bugs. Both unit testing and system testing are conducted to ensure the system works as expected.

- **Deployment**: After successful testing, the system is deployed for actual use. This may involve uploading it to a web server or making it available to users.

- **Release Product and Feedback**: Release the current version of the product and then take feedback and make changes in the new version.

**3.3 Modules**

The Online Complaint Portal is designed using a modular architecture that divides the overall functionality into logically independent components. Each module handles specific operations and responsibilities, working together to provide a seamless and efficient system.

The key modules in this project are as follows:

**User Management Module:**

This module handles authentication, role-based access control, and user profile management. There are three types of users:

* Admin: Can manage officers, view all complaints, generate reports, and access system settings.
* Officer: Handles assigned complaints, updates their status, and communicates with citizens.
* Citizen: Files complaints, tracks their status, and provides feedback on resolution.

Key Features:

* Secure login/logout functionality
* Password encryption and recovery
* Role assignment and access restriction
* Profile management and updates
* User activity logging

**Complaint Management Module:**

This is the core module of the system, allowing citizens to file complaints and officers to process them.

Key Features:

* Complaint submission with category selection
* File attachments for evidence/documentation
* Tracking system with unique complaint IDs
* Status updates with timestamps
* Officer assignment workflow
* Resolution documentation
* Feedback collection after resolution

**Officer Assignment Module:**

This module allows administrators to assign complaints to appropriate officers based on factors such as department, location, and current workload.

Key Features:

* Automatic or manual assignment options
* Officer availability tracking
* Workload balancing algorithms
* Reassignment capabilities
* Officer performance metrics
* Department and jurisdiction filtering

**Notification and Communication Module:**

This module facilitates communication between stakeholders and keeps everyone informed about important updates.

Key Features:

* Email notifications for new complaints, status changes, and assignments
* In-app messaging between officers and citizens
* Comment threads on complaints
* SMS alerts for critical updates
* Notification preferences management
* Automated reminders for pending actions

**Reporting and Analytics Module:**

Used primarily by administrators to generate insights and track system performance.

Key Features:

* Complaint volume analytics by category/location
* Resolution time metrics
* Officer performance reports
* Citizen satisfaction ratings
* Trend analysis over time periods
* Exportable reports in various formats (PDF, Excel)
* Interactive dashboards with visual data representation

**Administration and Settings Module:**

This module provides system-wide configuration options for administrators.

Key Features:

* Category and subcategory management
* System-wide announcement creation
* User account management
* Database backup and maintenance tools
* Email template customization
* System logs and audit trails
* Configuration of operational parameters

Each module is designed to be cohesive within itself while maintaining clear interfaces with other modules. This modular approach enhances maintainability, allows for parallel development, and makes the system more adaptable to future enhancements or modifications as requirements evolve.

**3.4 Analysis Model**

The analysis model for the Online Complaint Portal provides a comprehensive breakdown of the system's functionality and visualizes how it will work before moving into design and development stages. This model is crucial for understanding how different users interact with the system, what processes are involved, and how data flows between various components.

To clearly represent the system's behavior, several diagrams and models have been used:

**Use Case Diagram**

This diagram illustrates the main users—Admin, Officer, and Citizen—and their interactions with the system. It includes actions like logging in, filing complaints, assigning complaints to officers, tracking complaint status, generating reports, and managing user accounts. The use case diagram helps identify the primary functionalities needed for each user role.

**Activity Diagrams**

These diagrams illustrate the step-by-step flow of important processes within the complaint management system, such as:

* Filing a new complaint by citizens
* Officer assignment workflow by admins
* Complaint resolution process by officers
* Status updating and feedback collection

Activity diagrams clarify how the system handles user actions, decision points, and data flow within each role's operations, providing a clear visual representation of process workflows.

**Sequence Diagram**

The sequence diagrams show the step-by-step interactions between users and system components during specific processes. For example, the complaint submission sequence illustrates how a citizen's complaint moves through validation, storage, and notification processes. These diagrams focus on the time-based flow of messages, making them useful for modeling dynamic behaviors such as complaint submission, officer assignment, status updates, and dashboard data retrieval.

**Class Diagram**

The class diagram outlines the system's core objects like User, Citizen, Officer, Admin, Complaint, Category, Status, and Report, showing how they relate to one another. This provides the foundation for designing the database schema and establishing the system's object structure. The relationships between classes (such as "Citizen files Complaint" or "Officer resolves Complaint") are clearly defined to guide the implementation.

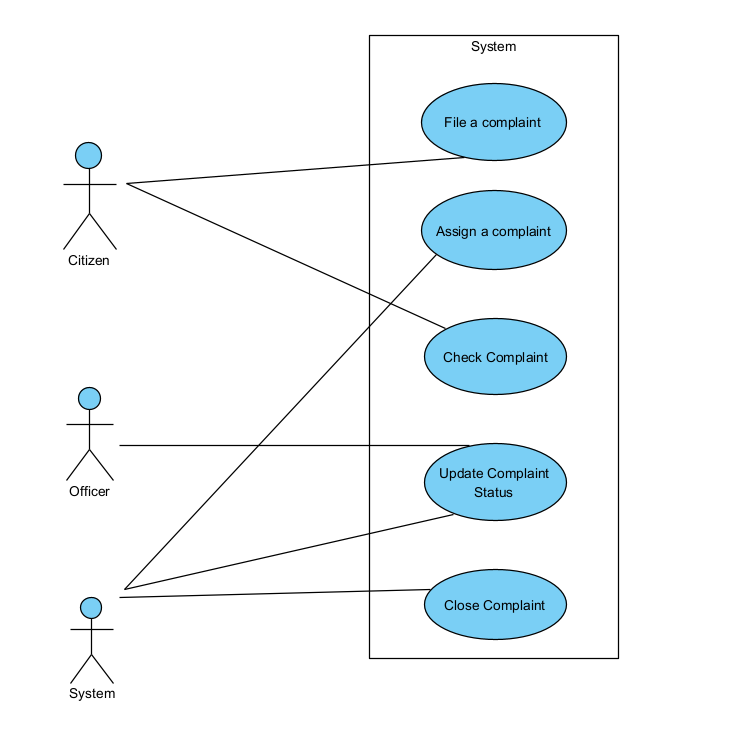
**Data Flow Diagrams (DFD)**

DFDs show how data moves through the complaint management system. They highlight:

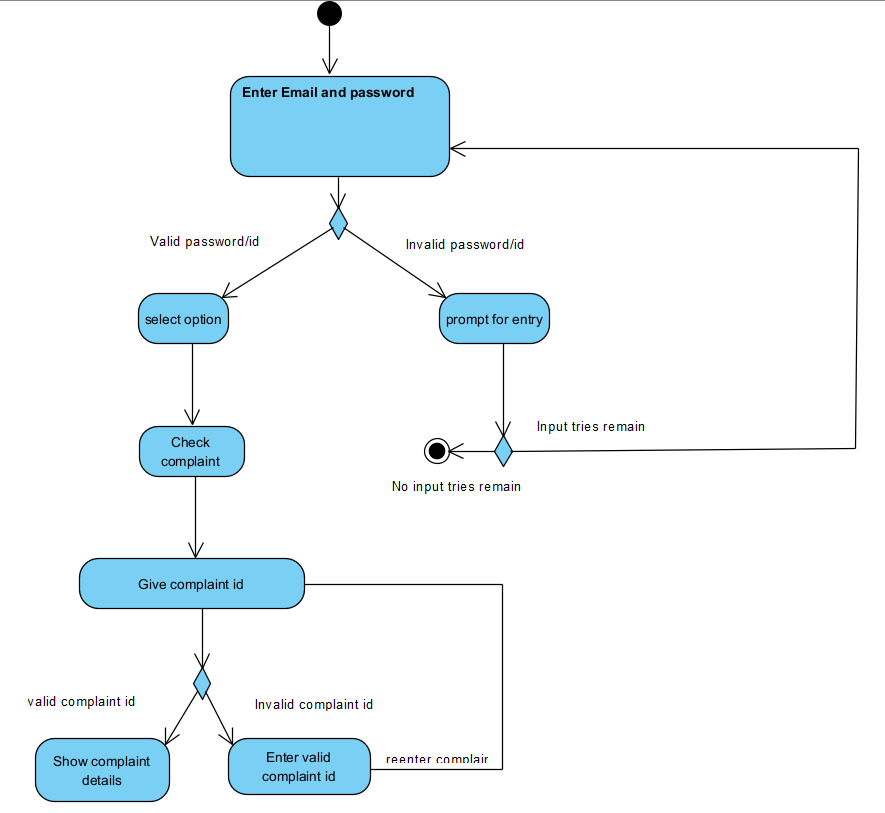
* Where data originates (e.g., complaint submission forms)
* How it is processed (e.g., complaint validation, assignment)
* Where it is stored (e.g., complaint database)
* How information flows to and from different users (citizens, officers, admins)

**3.5 UML Diagrams**

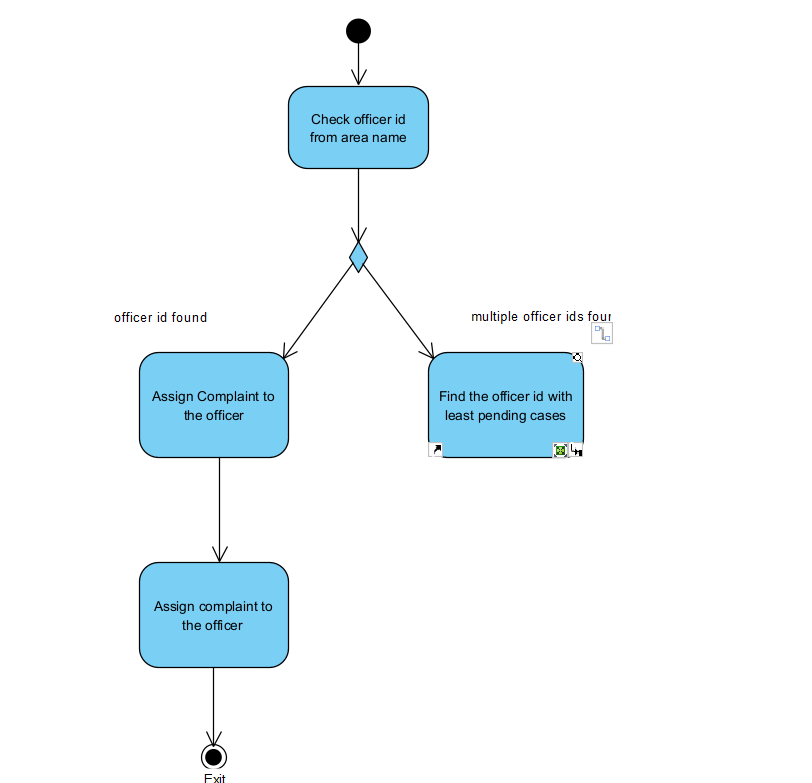
**3..1 Use Case Diagram :**

****

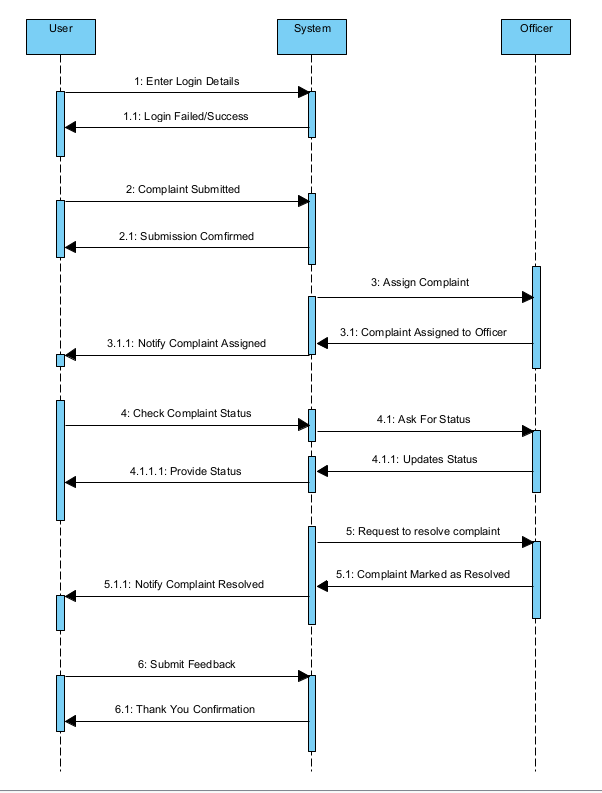
**3.5.2 Activity Diagram**

**3.5.2.1 For filing complaint**

**3.5.2.2 For Assigning Complaint**

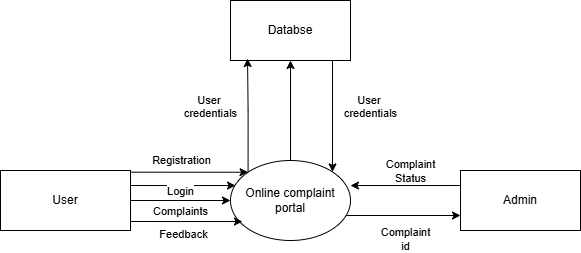
****

**3.5.3 Sequence Diagram**

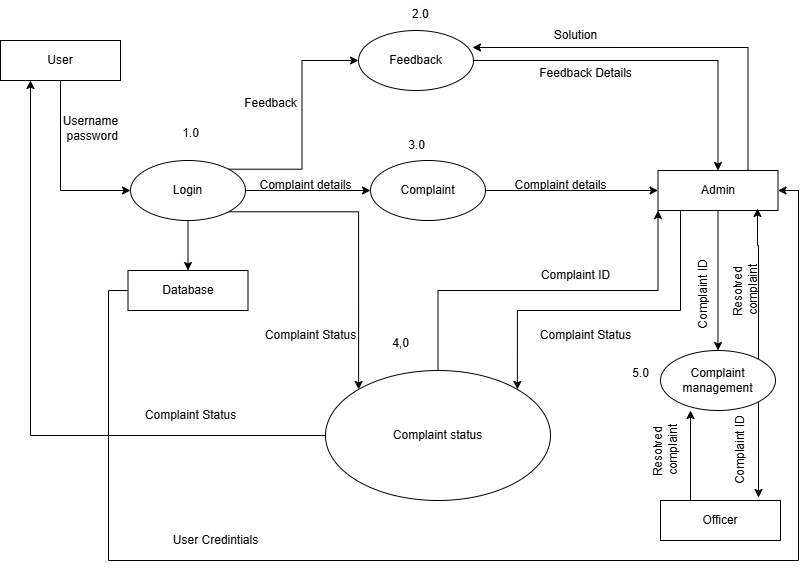
****

**3.5.4 DFD**

**Level-0**

****

**Level-1**



**CHAPTER-4**

**Technology Description & Architecture**

**4.1 System Architecture**

The architecture of the *Online Complaint Portal* follows a client-server model with a multi-layered design to ensure scalability, modularity, and maintainability. The system is organized into the following layers:

**4.1.1 Presentation Layer (Frontend)**

* Technologies Used: React and Tailwind CSS.
* Responsibilities:
  + Allows users (citizens and officers) to interact via forms and dashboards.
  + Provides user-friendly interfaces for complaint filing and tracking.
  + Validates form inputs before submission.
  + Displays real-time updates and complaint statuses.

**4.1.2 Application Layer (Backend)**

* Technologies Used: [Node.js](http://node.js) & Express
* Responsibilities:
  + Handles form submissions and server-side validations.
  + Manages user sessions, authentication, and role-based access.
  + Performs logic for complaint assignment, status updates, and notifications.
  + Routes data between frontend and database.

**4.1.3 Data Layer (Database)**

* Technology Used: MySQL
* Responsibilities:
  + Stores data for users, complaints, officer assignments, and feedback.
  + Maintains relationships between citizen IDs, officer IDs, and complaint records.
  + Ensures data consistency and supports structured queries.

**4.1.4 Notification Layer (Optional Enhancement)**

* Future Scope: SMS/Email APIs like Twilio or SMTP services
* Responsibilities:
  + Sends complaint status notifications to users.
  + Delivers registration confirmations or feedback alerts.

**4.2 Component Diagram Overview**

The major components of the system and their interactions are as follows:

1. User Interface: Citizens and officers interact via a web browser.
2. Web Server ([Node.js](http://node.js) & Express): Handles user requests and business logic.
3. Database (MySQL): Stores and retrieves user, complaint, and officer information.
4. Assignment Module: Matches complaints with appropriate officers based on area.
5. Status Tracker: Updates and displays complaint resolution progress.

Workflow Summary:

* User submits a complaint via the form.
* Backend verifies inputs and logs complaints into the database.
* System auto-assigns the complaint to an officer based on the area.
* Officer updates the complaint status (Pending → Resolved → Closed).
* Citizens can track complaints and confirm closure.

**4.3 Database and Schemas**

The *Online Complaint Portal* uses a relational database with multiple interlinked tables. Below are key schema designs:

**4.3.1 Citizens Table**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| citizen\_id | INT | Primary key |
| name | VARCHAR | Full name |
| username | VARCHAR | Unique login name |
| password | VARCHAR | Encrypted password |
| email | VARCHAR | For notifications |
| mobile | VARCHAR | Contact number |
| area\_id | INT | Foreign key linking to area |

**4.3.2 Officers Table**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| officer\_id | INT | Primary key |
| name | VARCHAR | Officer full name |
| username | VARCHAR | Unique login name |
| password | VARCHAR | Encrypted password |
| email | VARCHAR | Contact email |
| mobile | VARCHAR | Phone number |
| area\_id | INT | Foreign key to assigned jurisdiction |
| resolved\_complaints | INT | Total resolved complaint count |

**4.3.3 Complaints Table**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| complaint\_id | INT | Primary key |
| citizen\_id | INT | Foreign key to Citizens table |
| officer\_id | INT | Assigned officer |
| area\_id | INT | Complaint area |
| title | VARCHAR | Complaint subject |
| description | TEXT | Detailed message |
| photo\_path | VARCHAR | File path to uploaded image |
| status | ENUM | ('Pending', 'Resolved', 'Closed') |
| date\_submitted | DATETIME | Timestamp of complaint submission |

**4.3.4 Area Table**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| area\_id | INT | Primary key |
| area\_name | VARCHAR | Name of the area/region |

**4.4 Deployment Architecture**

* Hosting: Can be deployed on vercel.
* Database: MySQL hosted on the same or remote server.
* Storage: Complaint images stored in /uploads/ or cloud storage bucket(multer).
* Security:
  + Passwords hashed using bcrypt.
* Scalability: Designed for modular expansion (e.g., mobile app frontend).
* Monitoring: Admin panel with complaint and officer statistics for analytics and improvements.

**CHAPTER-5**

**RESULTS AND OUTPUTS**

**5.1 Functional Results**

The *Online Complaint Portal* has been successfully developed and tested to ensure its core functionalities operate as intended. The results of individual modules during implementation and testing are as follows:

* **User Registration & Login**:
  + Citizens and officers can register and authenticate using valid credentials.
  + Role-based redirection ensures correct dashboard access (citizen/officer).
* **Complaint Filing Module**:
  + Users can file complaints with details including area, description, and optional image upload.
  + Form validations work for required fields and acceptable file formats.
* **Auto-Assignment System**:
  + Complaints are assigned to officers based on area.
  + Load balancing mechanism assigned to officers with the least pending cases.
* **Complaint Tracking & Updates**:
  + Citizens can check complaint status using the complaint ID.
  + Officers can update status from “Pending” to “Resolved” or “Closed”.
  + A confirmation system allows citizens to verify resolution before closure.
* **Feedback System**:
  + Citizens can provide feedback after resolution confirmation.
  + Feedback entries are stored for administrative review.

**5.2 Sample Input-Output Observations**

| **User Role** | **Action** | **Input** | **System Response** |
| --- | --- | --- | --- |
| Citizen | File Complaint | Title: “Streetlight not working”  Area: 473226 | Complaint ID generated, status set to “Pending” |
| Officer | Update Complaint Status | Complaint ID: [OCP-511789-49](http://localhost:5173/complaints/18)  Status: Resolved | Status updated, citizen notified |
| Citizen | Check Complaint Status | Complaint ID:[OCP-511789-49](http://localhost:5173/complaints/18) | Status: Resolved displayed |
| Citizen | Submit Feedback | Rating: 4/5  Comment: “Quick action taken” | Feedback stored |

**5.3 Output Screens**

1. **User Dashboard**:
   * Displays complaint filing form for citizens.
   * Officers see assigned complaints with status options.
2. **Complaint History View**:
   * Citizens can view all complaints filed along with current status.
3. **Admin Panel** (Optional/Future Scope):
   * Overview of total complaints, resolution times, feedback summary.
4. **Notifications**:
   * Users receive alerts on status change or officer updates.
5. **Error & Validation Messages**:
   * Invalid complaint ID, missing fields, or duplicate usernames prompt appropriate warnings.

**5.4 Performance Metrics**

| **Metric** | **Observed Result** |
| --- | --- |
| Average Complaint Filing Time | ~3–4 seconds per submission (including image upload) |
| Complaint Status Update Time | ~2 seconds for officers |
| Database Query Time | < 1 second (optimized queries) |
| Accuracy of Area-based Assignment | 100% (as per test cases with known mappings) |
| Browser Compatibility | Fully functional on Chrome, Firefox, Edge |

**5.5 User Feedback Summary**

A user feedback survey was conducted among a group of test users (students and faculty) to evaluate the usability and performance of the system.

**Highlights:**

* **Ease of Use**: Most users found the portal simple and intuitive.
* **Transparency**: Users appreciated the ability to track complaint progress.
* **Responsiveness**: Officers reported smooth handling and status updates.

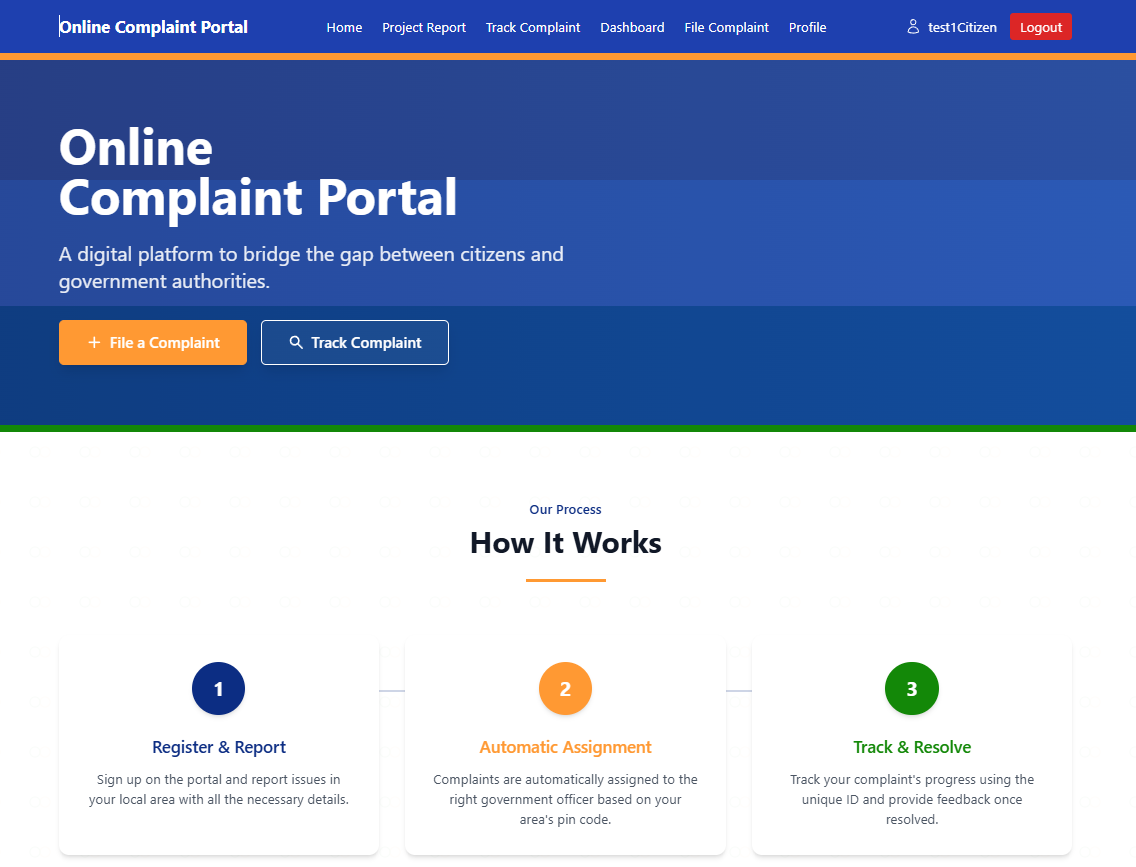
**Suggestions for Improvement:**

* Add SMS/email notifications on complaint updates.
* Support for multilingual complaint submission.
* Public view of resolved issues (optional transparency feature)

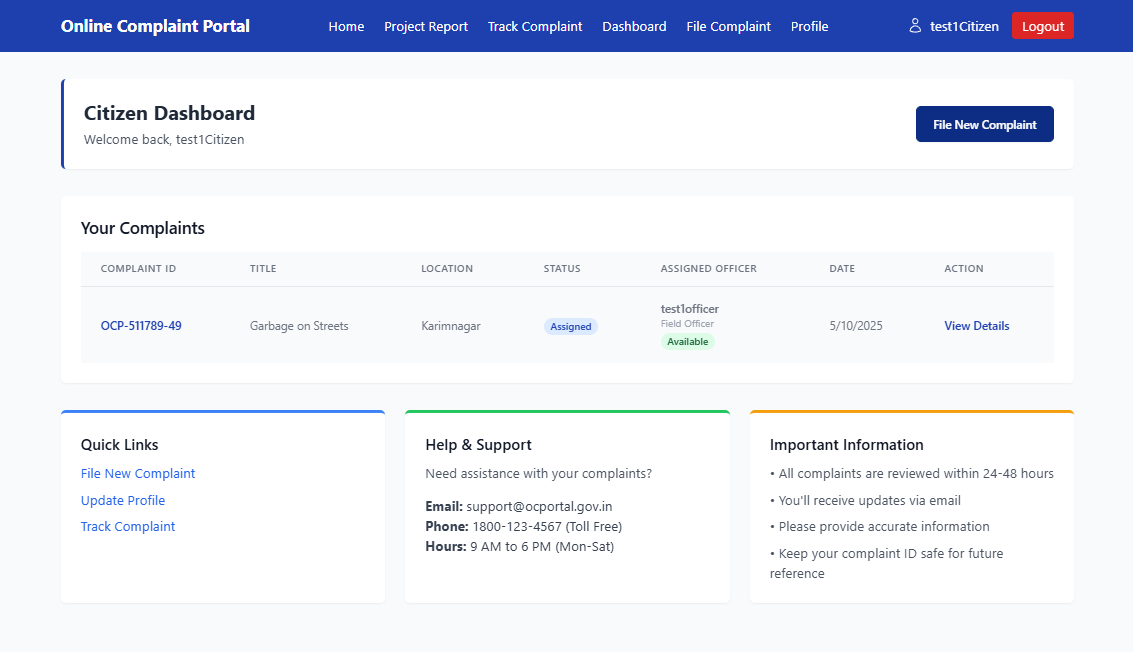
**5.6 Project Interfaces**

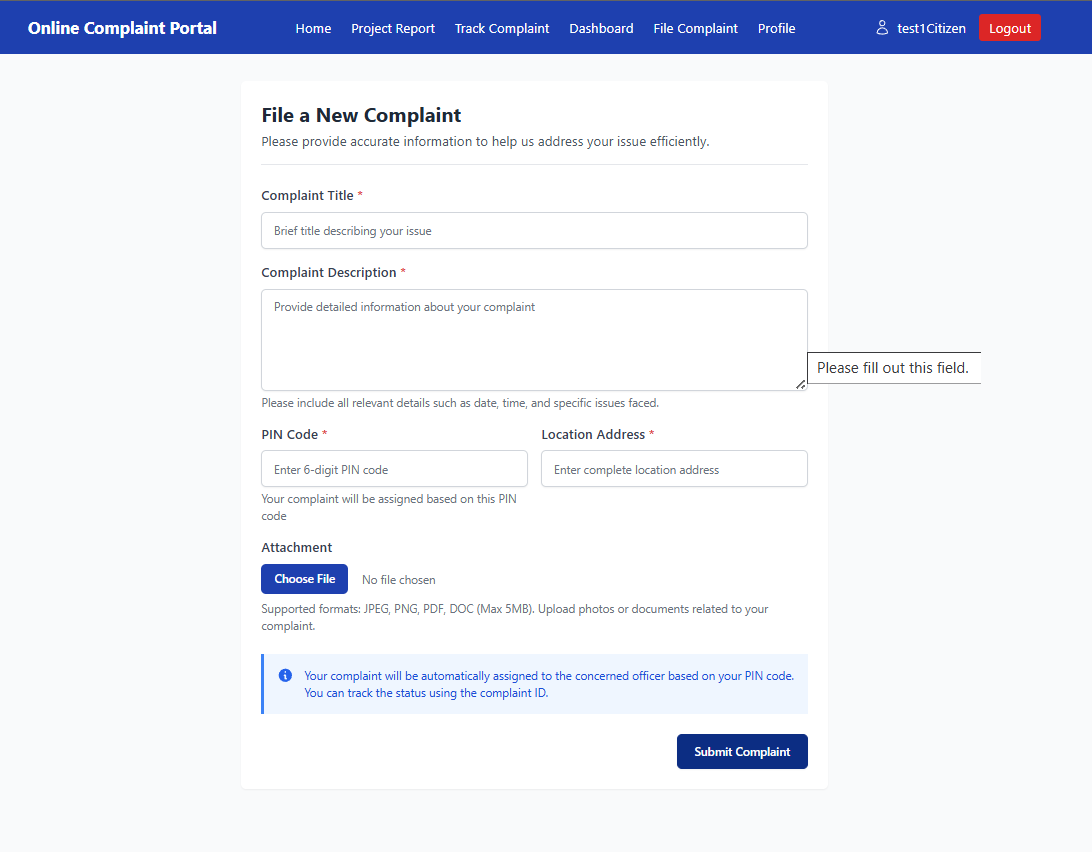
**5.6.1 Login/Registration page**

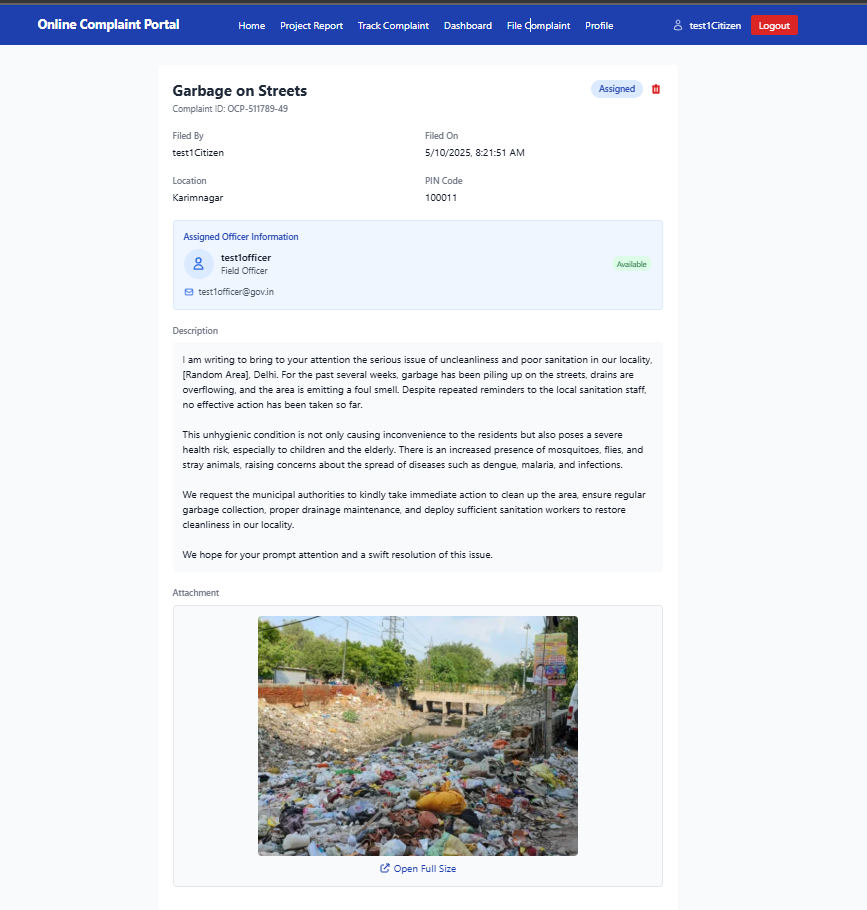
|  |
| --- |
|  |

**5.6.2 Home Page**

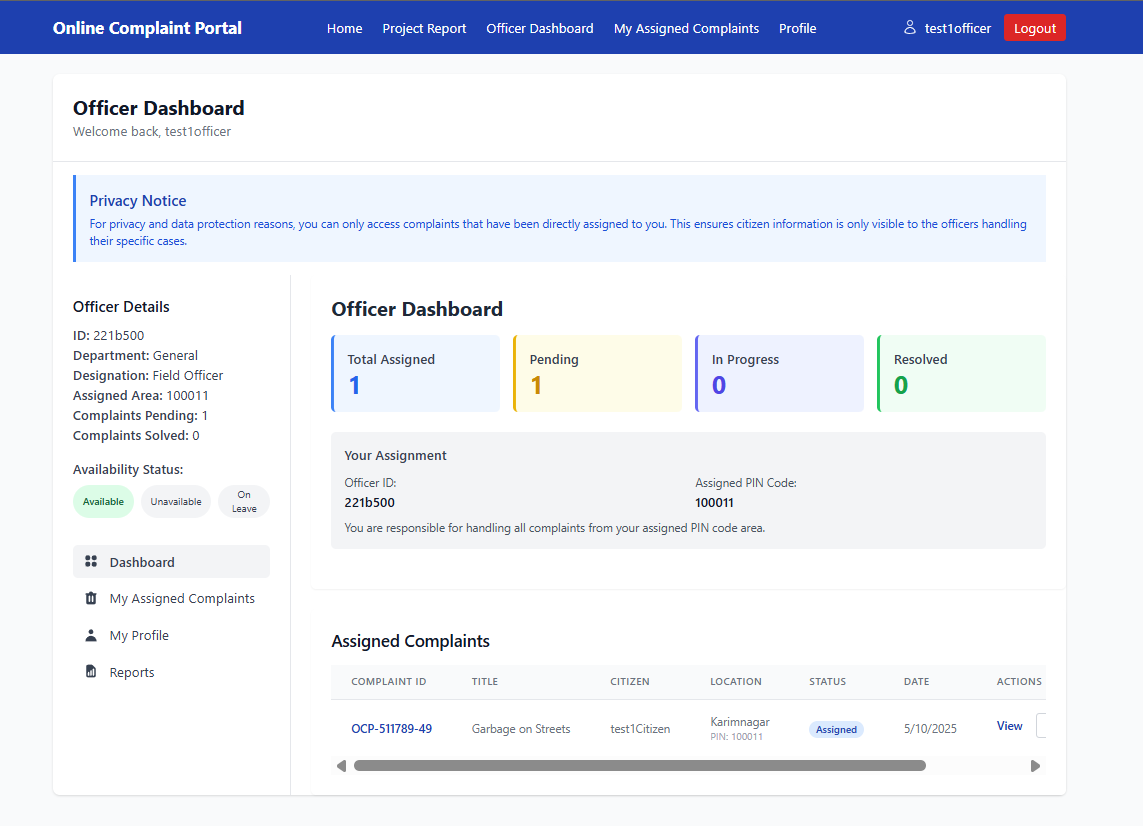
**5.6.3 Citizen Dashboard**

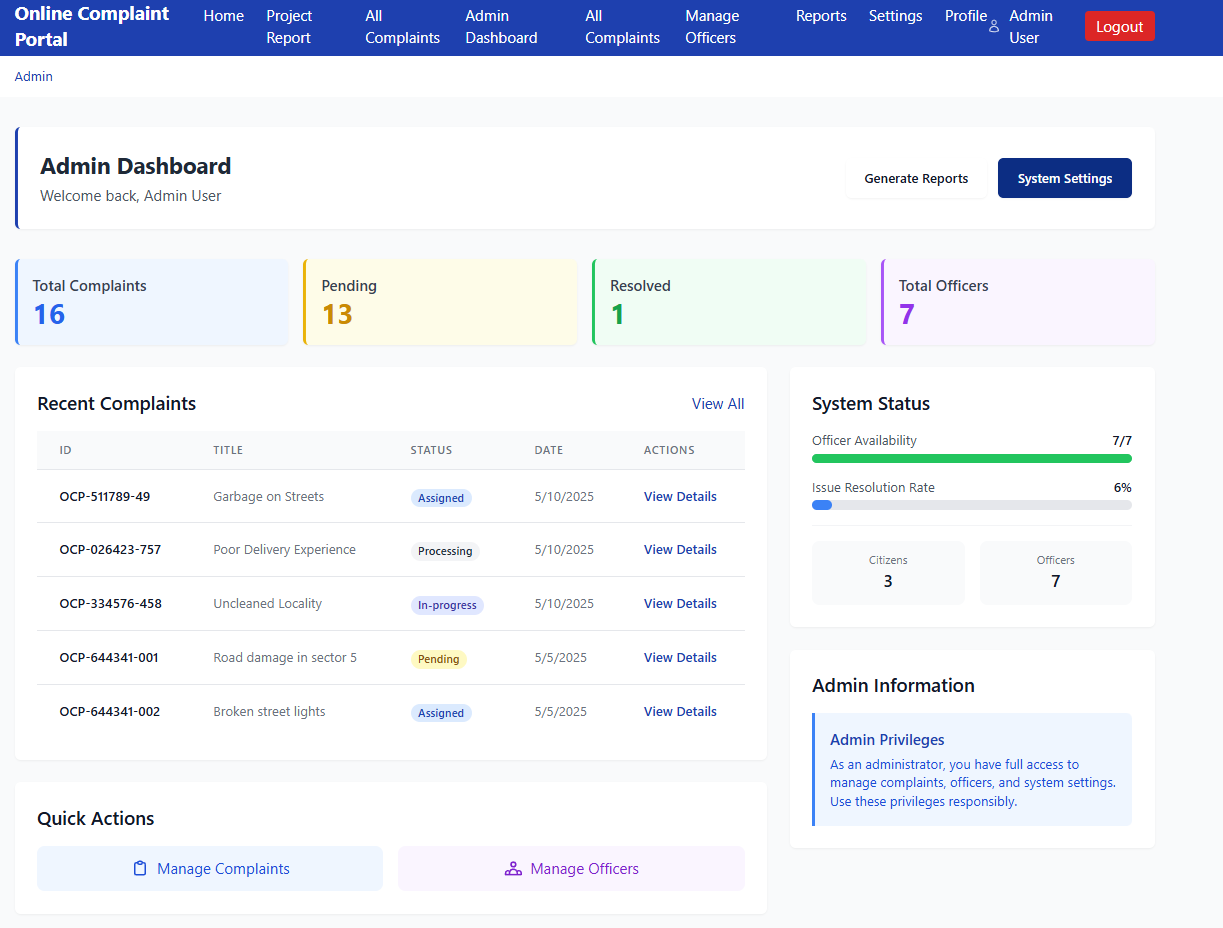
****

**5.6.4 File Complaint Page**

**5.6.5 View Complaint Page**

**5.6.6 Officer’s Dashboard Page**

****

**5.6.7 Admin’s Dashboard Page**

**CHAPTER-6**

**CONCLUSION**

The *Online Complaint Portal* project successfully addresses the common challenges in traditional grievance redressal systems by offering a transparent, efficient, and user-friendly digital solution. Through the implementation of modern web technologies and a modular architecture, the portal enables citizens to file complaints, track their status in real-time, and receive timely resolutions from designated government officers.

The system ensures:

* A significant reduction in manual paperwork and physical visits to government offices.
* Improved communication and accountability between public users and government entities.
* Centralized data storage for complaints, facilitating easy tracking and analytics.
* Role-based access control to separate citizen and officer functionalities securely.

Throughout the project, special attention was given to usability, performance, and scalability. The system was tested across various modules such as registration, complaint filing, assignment, tracking, and feedback, with all core functions operating successfully.

This project serves as a foundation for further enhancements such as:

* Integration with mobile applications.
* Multi-language support.
* Real-time notifications via SMS or email.
* Advanced analytics dashboards for administrators.

In conclusion, the *Online Complaint Portal* is a scalable and adaptable platform that can be implemented across various government departments, educational institutions, and private organizations to ensure streamlined and transparent handling of grievances. It promotes citizen empowerment, operational efficiency, and trust in public service delivery.

**CHAPTER-7**

**REFERENCE**

 <https://www.tutorialsduniya.com> – Source for software engineering project templates and documentation examples.

 <https://www.w3schools.com> – Reference for React, Tailwind CSS, Node.js, and Express.js.

 <https://www.nodejs.org> – Official Node documentation.

 <https://dev.mysql.com/doc> – Official MySQL documentation.

 <https://draw.io> – Online tool used to design the use case and architecture diagrams.