CIVIC VOICE

A Mini Project Report

submitted by

JASNA SHERIN M

(MES24MCA-2024)

to the APJ Abdul Kalam Technological University
in partial fulfilment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering

Kuttippuram, Malappuram – 679582

October, 2025

Declaration

I undersigned hereby declare that the project report CIVIC VOICE submitted for partial fulfilment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bona fide work done by me under the supervision of Mr. Vasudevan T V, Assistant Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

JASNA SHERIN M

(MES24MCA-2024)

06/08/25

DEPARTMENT OF COMPUTER APPLICATIONS

MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **CIVIC VOICE** is a bonafide record of the Mini Project work during the year 2025-26 carried out by **JASNA SHERIN M (MES24MCA-2026)** submitted to the APJ Abdul Kalam Technological University, in partial fulfilment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor   Head of The Department

Acknowledgment

I would like to begin by expressing my deep sense of gratitude to the Almighty, whose blessings and guidance have been a constant source of strength and inspiration throughout the completion of my project work titled “CIVIC VOICE”, undertaken as part of the requirements for the degree of Master of Computer Applications at MES College of Engineering, Kuttipuram, under APJ Abdul Kalam Technological University, Kerala.

I owe my sincere thanks to my project guide, Mr. Vasudevan T V, Assistant Professor, Professor Hyderali K Head of The Department of Computer Applications,MES College of Engineering, Kuttipuram. for his valuable guidance, constructive suggestions, and constant encouragement at every stage of the project. His insights and expertise have been instrumental in shaping this work into its final form. I also express my heartfelt gratitude to the Head of the Department of Computer Applications for providing me with the necessary facilities, encouragement, and academic support during the course of my project. My sincere thanks also go to all the faculty members and technical staff of the Department of Computer Applications for their timely advice, cooperation, and moral support. I am especially thankful to my friends and classmates for their valuable feedback, collaboration, and encouragement, which motivated me to enhance the quality of this work.

Finally, I express my deepest gratitude to my family members for their unconditional love, patience, and constant support, which gave me the confidence to successfully complete this project.

JASNA SHERIN M

(MES24MCA-2024)

Abstract

CivicVoice is a digital platform designed to streamline the process of public grievance redressal by enabling citizens to file complaints related to various government services through an organized and centralized online portal. With public services such as education, healthcare, transport, and sanitation playing a vital role in everyday life, the need for an accessible and transparent complaint management system has become essential. Traditional methods of lodging complaints are often inefficient, slow, and lack accountability. CivicVoice addresses these challenges by offering a structured, user-friendly interface that allows users to register complaints, categorize them by sector, and track their resolution status.

At its core, the system leverages a robust database and automated complaint routing mechanism that ensures each grievance is directed to the appropriate government department based on its category and location. By assigning a unique complaint ID, the system allows users to monitor progress and receive timely status updates in the system . A secure login system supports user account management, while the administrative dashboard facilitates complaint review, action tracking, and record maintenance by authorized officials. This structured approach promotes accountability and efficiency in the resolution process.

The platform emphasizes data integrity, administrative transparency, and system scalability. With features such as complaint history logs, acknowledgment emails, and real-time tracking, CivicVoice not only empowers citizens to voice their concerns but also enables government authorities to manage and resolve issues systematically. Developed using Python (Django), HTML, CSS, JavaScript, and backed by database SQLite, the platform ensures cross-platform compatibility and ease of maintenance. Tools such as Git and Visual Studio Code support collaborative development and version control.

Overall, CivicVoice serves as a bridge between citizens and public service providers, facilitating effective governance and improving public trust in government systems. By digitizing and streamlining the grievance redressal process, it contributes to a more responsive and citizen-centric administration.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

The rapid advancement of digital technologies and e-governance systems has transformed the way public services are delivered and managed. In today’s world, citizens expect fast, transparent, and accessible platforms to interact with government departments and resolve their concerns. However, the traditional grievance redressal mechanisms — such as physical complaint registers, helplines, or department visits — are often inefficient, time-consuming, and lack accountability. This results in delays, data loss, and poor communication between citizens and administrative bodies.

The project “Civic Voice” aims to overcome these challenges by providing a centralized, user-friendly digital platform for public grievance redressal. The system enables citizens to file complaints related to various government sectors such as education, healthcare, transport, and sanitation through an organized online interface. It also ensures that each complaint is automatically routed to the relevant government department based on its category and location, eliminating manual forwarding delays.

Built using Python (Django) as the backend framework and HTML, CSS, and JavaScript for the frontend, the platform integrates secure user authentication, automated routing, and real-time complaint tracking. Each grievance is assigned a unique complaint ID, enabling users to monitor the progress and receive updates via email or SMS. On the administrative side, the system provides a robust dashboard that allows officials to review, update, and close complaints efficiently. The database, designed using SQLite, ensures reliability, scalability, and easy maintenance.

1.1 Motivation

The primary motivation behind developing Civic Voice is to strengthen the relationship between citizens and government agencies by improving the efficiency and accountability of grievance redressal processes. Traditional methods such as complaint boxes, office visits, or helplines are slow and inconsistent. Often, citizens are unaware of the status of their complaints, and departments face difficulties in tracking and analyzing large volumes of submissions. From a citizen's perspective, there is a strong need for a transparent, real-time tracking system where grievances can be lodged quickly and monitored until resolved. From a government perspective, an organized digital platform ensures proper complaint categorization, better workload distribution, and data-driven insights for policy improvement. Academically, the project offers an opportunity to apply modern web technologies such as Python (Django) and relational databases to solve a real-world governance problem. It demonstrates how structured system design can improve administrative workflows while promoting civic engagement and accountability

1.2 Objectives

The main objective of this mini project is to design and develop a web-based platform that enables citizens to register and monitor grievances effectively while allowing administrators to process and resolve them in a timely manner. The specific objectives of Civic Voice are as follows:

1. To develop an online complaint registration system with secure citizen login and authentication.
2. To implement automatic complaint routing based on category and location to the relevant government department.
3. To enable real-time complaint tracking using a unique complaint ID and provide status updates via email notification.

4.To create an administrative dashboard for monitoring, updating, and resolving complaints efficiently.

5.To maintain a centralized database of complaints and actions taken for transparency and accountability.

6.To promote accessibility and citizen empowerment through a simple, user-friendly interface.

1.3 Contributions

The major contributions of **Civic Voice** can be summarized as follows:

- **Unified Platform:** Provides a single digital portal for citizens to file and track grievances related to multiple public departments.
- **Automatic Complaint Routing:** Uses a logic-based routing mechanism to direct complaints to the correct department automatically.
- **Real-Time Tracking:** Offers live updates on complaint status through unique complaint IDs and email alerts.
- **Administrative Dashboard:** Enables officials to manage, prioritize, and close complaints systematically.
- **Transparency & Accountability:** Maintains a complete digital history of every complaint, promoting responsible governance.
- **Data Analytics:** Allows generation of reports for performance monitoring and policy improvement.
- **User-Friendly Interface:** Designed to be intuitive and accessible, ensuring easy use even for non-technical users.

1.4 Report Organization

The project report is organized into five chapters:

- **Chapter 1: Introduction** – Provides the background, motivation, objectives, contributions, and organization of the report.
- **Chapter 2: System Study** – Describes the existing systems and their limitations, followed by details of the proposed **Civic Voice** system and its functionalities.
- **Chapter 3: Methodology** – Explains the methodology used for implementation, including design approach, software tools, and detailed module descriptions. Sprint details and database design are also included.
- **Chapter 4: Results and Discussions** – Presents implementation results, screenshots of the developed system, and evaluation of performance with respect to objectives.
- **Chapter 5: Conclusion and Future Work** – Summarizes key outcomes, highlights contributions, and suggests directions for future improvement

Chapter 2. System Study

The project under consideration, Civic Voice, addresses the need for a centralized, transparent, and efficient public grievance redressal system. The goal is to simplify the process of lodging and managing complaints related to government services such as education, healthcare, transport, and sanitation. The system enables citizens to register complaints easily and track their progress in real time while providing administrators with tools to manage, update, and resolve grievances systematically. Developed using Python (Django) and SQLite. the system ensures scalability, security, and reliability for both users and administrators.

2.1 Existing System

In the current scenario, citizens mostly rely on traditional methods such as visiting government offices, using complaint boxes, or contacting helplines to register their grievances. Some online platforms like CPGRAMS and mSeva exist, but they are limited to specific departments or regions and often require manual forwarding of complaints. This makes the process slow, confusing, and inconvenient for users.

Because there is no unified system to handle all types of public complaints, tracking their progress or ensuring accountability becomes difficult. Citizens rarely receive timely updates about their complaints, and departments struggle to manage and analyze large volumes of submissions efficiently. These drawbacks highlight the need for a centralized, automated, and transparent grievance redressal system like Civic Voice.

2.2 Proposed System

The proposed system, Civic Voice, is a web-based grievance redressal platform that provides a centralized and automated way for citizens to submit and track complaints related to government services. When a user registers a complaint, the system automatically routes it to the appropriate department based on the selected category and location. Each complaint receives a unique complaint ID that allows the user to monitor its progress and receive regular updates through email or SMS notifications.

The administrative side of the system offers a secure dashboard where officials can view, filter, and manage complaints efficiently. They can update the complaint status as pending, in progress, resolved, or closed, and generate performance reports for review and analysis. The database stores all complaint records, ensuring transparency and easy retrieval of information when required. By automating the complaint-handling process, Civic Voice eliminates delays caused by manual forwarding and improves communication between citizens and authorities. The system enhances transparency through real-time tracking, promotes accountability by maintaining digital complaint histories, and increases efficiency through organized workflows and data-driven insights. In this way, Civic Voice creates a more responsive and citizen-friendly

2.3 Functionalities of Proposed System

- **User Registration and Login**

Citizens can create an account and log in securely to submit and track their complaints.

- **Complaint Submission**

Users can file complaints by selecting the category, location, and description of the issue.

- **Automatic Routing**

The system automatically sends each complaint to the correct department based on the category and area selected.

- **Real-Time Tracking**

Every complaint is given a unique ID, allowing users to check its current status at any time.

- **History Complaint**

Users can view all their past complaints and the actions taken on them.

- **Admin Dashboard**

Officials can view and manage complaints, update their progress, and mark them as resolved or closed.

- **Report Generation**

Administrators can create reports to analyze complaints and improve department performance.

- **Feedback**

Admins can send replies or ask for more details from citizens to ensure better communication.

- **Data Storage**

All complaints and updates are stored safely in a database for transparency and future use.

Chapter 3.Methodology

Developing a web-based grievance redressal platform like Civic Voice requires the use of a systematic and efficient software development methodology to ensure timely completion, accuracy, and quality. The chosen methodology provides a structured approach for planning, designing, developing, testing, and deploying the application. Since the project involves dynamic features such as complaint submission, automated routing, real-time tracking, and feedback mechanisms, the Agile methodology was adopted to ensure flexibility, continuous improvement, and user satisfaction throughout the development process.

3.1 Introduction

The development of Civic Voice follows an agile and iterative approach to ensure flexibility, continuous feedback, and user-centered design. The system was designed to streamline the process of grievance redressal by dividing the project into smaller, manageable phases called sprints. Each sprint focuses on specific modules such as user authentication, complaint submission, tracking, and administrative management.

The agile methodology ensures close interaction between the developer and stakeholders, allowing frequent testing, feedback, and refinement. This approach improves software quality, reduces risks, and ensures that the final system meets both functional and non-functional requirements effectively.

3.2 Software Tools

The development of **Civic Voice** was carried out using open-source technologies and tools suitable for full-stack web development. The list of tools used for development is shown below.

Table 3.1: Software Tools Used for Development

Operating System	Windows 11
Front End	JavaScript, HTML, CSS etc..
Back End	Python
Framework	Django
Database	SQLite
IDE	Visual Studio Code
Version Control	Git

3.2.1 Python

Python was chosen as the backend programming language because of its simplicity, readability, and support for rapid web application development. It provides strong integration with databases and supports a variety of libraries for web handling, data management, and security implementation.

3.2.2 Django

Django is a high-level Python web framework that encourages rapid development and clean design. It was selected for Civic Voice because it includes built-in features such as authentication, ORM (Object Relational Mapping), and admin interface, which help in quick implementation of core functionalities. Django also ensures security, scalability, and maintainability of the application.

- **Front End (HTML, CSS, JavaScript)**

HTML, CSS, and JavaScript were used to design the front end of the Civic Voice platform. HTML structures the web pages, CSS is used for styling and layout, and JavaScript adds interactivity and dynamic content. Together, they create a responsive and user-friendly interface that allows smooth navigation and complaint submission.

- **Back End (Python)**

Python serves as the core backend language due to its simplicity, flexibility, and wide support for web application development. It provides built-in libraries for data handling, form processing, and validation. Python also integrates efficiently with Django to handle requests, route data, and manage security.

- **Framework (Django)**

Django was chosen as the primary web framework because it simplifies backend development through its modular structure and powerful ORM (Object Relational Mapping). It handles user authentication, database queries, and admin interface generation automatically. Django's built-in security mechanisms help protect user data and prevent threats such as SQL injection and cross-site scripting.

- **Database (SQLite)**

The project uses SQLite during development and MySQL for deployment, ensuring scalability and reliability. The database stores user details, complaint records, and departmental responses. It maintains data consistency and supports complex queries for generating reports and tracking complaint statuses.

- **IDE (Visual Studio Code)**

Visual Studio Code is used as the main Integrated Development Environment (IDE) for writing, testing, and debugging code. It provides useful extensions for Python, Django, and HTML, along with features like syntax highlighting and integrated Git support.

- **Version Control (GitHub)**

GitHub is used for version control and collaborative development. It allows tracking of code changes, managing different versions, and backing up the project in a secure online repository. GitHub also helps in maintaining transparency in the development workflow and facilitates teamwork.

3.3 Module Description

A software system is often divided into smaller manageable components called modules. Each module is designed to handle a specific part of the overall functionality of the system. This modular approach improves maintainability, scalability, and clarity of the project. By separating responsibilities, developers can work on different parts of the system independently, which also simplifies debugging, testing, and future enhancements.

The **Civic Voice** application consists of two main modules:

1. **Citizen Module** – Focused on user-related functionalities such as registration, complaint submission, status tracking, and feedback.
2. **Admin Module** – Designed for administrators to manage complaints, update statuses, generate reports, and ensure efficient redressal of issues.

3.3.1 Citizen Module

The Citizen Module handles all user-side interactions and provides an accessible and efficient platform for grievance redressal. Citizens can securely register and log in using their credentials, ensuring safe access to the system. Once authenticated, users can lodge complaints by selecting the relevant department or service category, specify their location, and describe their issue in detail. After submission, each complaint is automatically assigned a unique complaint ID, which the citizen can use to track its progress in real time.

The module also allows users to view their complaint history, including previously submitted issues and their resolution status. It provides a user-friendly interface developed using HTML, CSS, and JavaScript, ensuring that even first-time users can navigate the system without difficulty. The Citizen Module improves transparency and trust by allowing users to actively participate in governance through digital complaint management and real-time tracking.

3.3.2 Admin Module

The Admin Module enables government officials and authorized personnel to manage, monitor, and resolve citizen complaints effectively. This module provides a secure login system for administrators, ensuring that only authorized users can access departmental data.

Once logged in, administrators can view all complaints assigned to their department, categorized by their current status — Pending, In Progress, Resolved, or Closed. They can also update complaint details, add remarks, and change the status as actions are taken. The system automatically logs all updates, which ensures accountability and accurate record-keeping.

The Admin Module includes a dashboard interface that provides quick access to complaint summaries and analytical reports. Officials can generate reports based on date, department, or complaint type to evaluate performance and identify recurring issues. Additionally, administrators can communicate with citizens through in-system feedback or messages if more information is required.

This module ensures transparency, faster issue resolution, and better communication between citizens and officials. By maintaining digital records and automating processes, the Admin Module enhances operational efficiency and contributes to improved governance.

3.4 User Story

A user story represents a short and simple description of a specific feature from the perspective of the end user. In the Civic Voice system, each user story captures what the user wants to achieve and why it is important. This helps define the system's requirements from a practical point of view. For example, a citizen might want to register a complaint so that they can raise issues related to public services, or an administrator might want to update the complaint status to ensure timely redressal. These user stories help identify the core functionalities of the system and ensure that development aligns with the real needs of both citizens and administrators

Table 3.2 : User Story

User Story ID	As a type of User	I want to	So that I can
1	Citizen	Register	Create an account and gain access to the complaint portal.
		Login	Access my account securely with valid credentials.
		Submit Complaint	Report issues related to public services such as health, education, or sanitation.
		Track Complaint	Check the current status and progress of my submitted complaints.
		View Complaint History	Review previous complaints and their resolution details.
		Provide Feedback	Share my experience after the complaint is resolved.

User Story ID	As a type of User	I want to	So that I can
2	Admin	Login	Access the system dashboard securely for authorized use.
		View Assigned Complaints	See all complaints assigned to my department for action.
		Update Complaint Status	Mark complaints as Pending, In Progress, Resolved, or Closed with remarks.
		Report Analytics	Analyze department performance and complaint statistics.
		Communicate with Citizens	Request additional information or provide responses to users.

3.5 Product Backlog

A product backlog is a prioritized list of all features, functionalities, and tasks that need to be developed in the project. It serves as a central reference for managing and tracking development progress. The backlog is continuously updated throughout the project based on feedback, testing, and implementation priorities. In the Civic Voice system, the product backlog includes the main features required to build a complete and functional grievance redressal platform. The features are categorized and prioritized according to their importance for ensuring user satisfaction, transparency, and efficient system performance.

Table 3.3: Product Backlog

ID	**Feature**	**Priority**	**Estimated Hours**	**Status**
1	User registration and login	High	5 hrs	Completed
2	Complaint submission	High	6 hrs.	Completed
3	Automatic complaint routing	High	5 hrs.	Completed
4	Complaint tracking	High	4 hrs.	Completed
5	Complaint history view	Medium	3 hrs.	Completed
6	Admin login and authentication	High	4 hrs.	Completed
7	Admin dashboard for complaint management	High	6 hrs	Completed
8	Complaint status update and remarks	High	5 hrs	Completed
9	Report Analytics by admin	Medium	4 hrs	Completed

10	Feedback module	Medium	3 hrs	Completed
11	Database integration	High	3 hrs	Completed
12	Testing and debugging	Medium	2 hrs	Completed

3.6 Project Plan

A project plan provides an organized schedule for completing the various stages of development within the given time frame. It helps ensure that each phase of the project is completed systematically and on time. The plan also defines the order of tasks, estimated duration, and current progress status for each phase. The Civic Voice project was developed using the Agile methodology, where the work was divided into multiple sprints. Each sprint focused on a specific set of functionalities such as user registration, complaint submission, routing, and admin management. The following table outlines the key phases and activities of the project plan.

Table 3.4: project plan

User StoryID	**Task Name**	**Start Date**	**End Date**	**Days**	**Status**
1	Sprint 1	18/08/2025	21/08/2025	20	Completed
1		22/08/2025	25/08/2025		Completed

1		26/08/2025	08/09/2025		Completed
1	Sprint 2	09/09/2025	11/09/2025	11	Completed
1		12/09/2025	19/09/2025		Completed
1	Sprint 3	20/09/2025	25/09/2025	7	Completed
2		24/09/2025	27/09/2025		Completed
2	Sprint 4	28/09/2025	08/10/2025	16	Completed
2		09/10/25	12/10/2025		Completed

3.7 Sprint Backlog

The sprint backlog represents the detailed task-level breakdown for each sprint during the development of the Civic Voice project. Each sprint focuses on a specific group of functionalities derived from the product backlog. The tasks are distributed according to user stories, with the duration, status, and time allocation for each activity clearly defined. The backlog was updated regularly to monitor progress and ensure smooth completion of both Citizen and Admin modules. The development work was carried out through two main sprints.

Table 3.5: sprint backlog

Backlog item	**Status And Completion Date**	**Original Estimation in Hours**	**Day 1 hrs**	**Day 2 hrs**	**Day 3 hrs**	**Day 4 hrs**	**Day 5 hrs**	**Day 6 hrs**	**Day 7 hrs**	**Day 8 Hrs**	**Day 9 hrs**	**Day 10 hrs**
SPRINT1												
User registration& login	18/08/2025	4	1	1	1	1	0	0	0	0	0	0
Complaint submission form	22/08/2025	5	1	1	1	1	1	0	0	0	0	0

Complaint categorization logic	26/08/2025	4	1	1	1	1	0	0	0	0	0	0
` SPRINT 2												
Complaint Routing with ID	09/09/2025	10	1	1	1	1	1	1	1	1	1	1
Tracking System	12/09/2025	5	1	1	1	1	1	0	0	0	0	0
SPRINT 3												
Admin Dashboard	20/09/2025	7	1	1	1	1	1	1	1	0	0	0

Complaint Status Update module	24/09/2025	6	1	1	1	1	1	1	0	0	0	0
SPRINT 4												
Reports analytics	28/09/2025	5	1	1	1	1	1	0	0	0	0	0
Feedback system	09/10/2025	4	1	1	1	1	0	0	0	0	0	0
TOTAL		50	9	9	9	9	5	6	2	1		

3.8 Database Design

A well-structured database design plays a crucial role in ensuring the efficiency, reliability, and scalability of any web application. For the Civic Voice system, the database is designed to store and manage all information related to users, complaints, departments, and administrative actions.

The database ensures proper relationships among entities, minimizes redundancy, and maintains data integrity. The system uses a relational database model implemented in SQLite, which is lightweight and highly compatible with Django.

Table 3.6: Database Design

Collection	Attributes	Purpose
Users	•user_id(PK) •name •email •password •phone •address • created_at	Stores details of registered citizens for authentication, profile management, and communication.
Departments	•dept_id(PK) •dept_name •dept_email •dept_phone • created_at	Maintains information about different public service departments such as Health, Transport, and Sanitation.
Admins	•admin_id(PK) •dept_id(FK) •name •email •password •role • created_at	Stores administrator credentials and their departmental assignments for complaint handling and system control.
Complaints	•complaint_id(PK) •user_id(FK) •dept_id(FK) •title •description •category •location •priority •status_id(FK) •created_at • updated_at	Records user-submitted grievances with all details like issue type, location, and status for tracking and resolution.

complaint_status	•status_id(PK) •status_name • created_at	Maintains standard workflow states for each complaint such as Pending, In Progress, Resolved, and Closed.
complaint_history	•history_id(PK) •complaint_id(FK) •admin_id(FK) •from_status •to_status •remarks • changed_at	Logs every status change of complaints to maintain transparency and accountability.
Feedback	•feedback_id(PK) •user_id(FK) •complaint_id(FK) •rating •comments • created_at	Stores user feedback on complaint resolution for performance evaluation and service improvement.
Attachments	•attachment_id(PK) •complaint_id(FK) •file_name •file_path • uploaded_at	Holds file details (images or documents) uploaded as evidence or references for complaints.

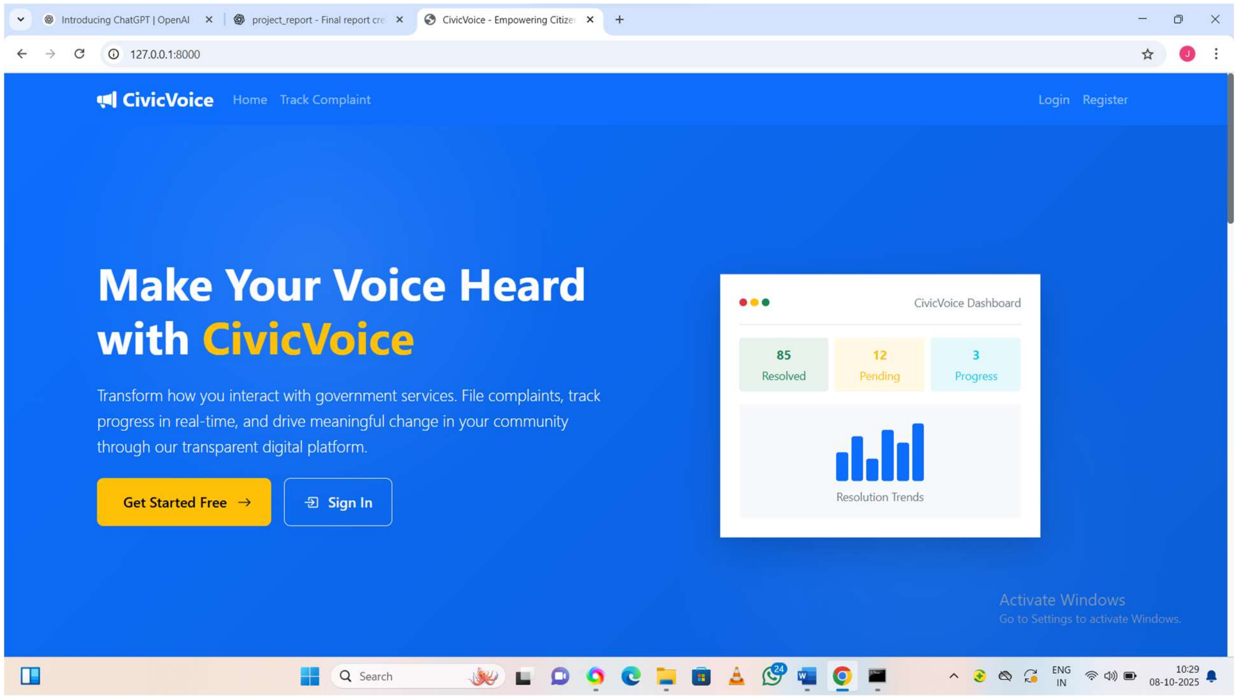
Chapter 4 . Results and Discussions

This chapter presents the outcomes of implementing the Civic Voice application and explains how the developed modules meet the objectives defined earlier. The system was designed and implemented using the Django framework with SQLite as the backend. The results show the successful development of major functionalities such as citizen registration and login, complaint submission, automatic routing to departments, status tracking, and feedback collection. Screenshots of the application are provided to illustrate the working of each module.The Citizen Module allows users to easily register complaints and track their progress through a simple and interactive interface. Each complaint is assigned a unique ID that helps users follow updates until resolution. The Admin Module enables authorized officials to view complaints, update their status, add remarks, and generate reports. This ensures that all issues are handled systematically and transparently.

Overall, the implementation results show that the Civic Voice system achieves its goal of improving communication between citizens and public authorities. It reduces manual paperwork, increases accountability, and enhances transparency in the grievance redressal process. The developed platform demonstrates a reliable and efficient digital approach to handling public complaints.

4.1 Results

This section highlights the core working modules of the Civic Voice system and demonstrates how each implemented functionality performs in practice. The results are presented with screenshots to show the user interface and workflow of both citizen and admin operations. The developed system successfully enables users to register complaints, track their status, and provide feedback, while administrators can review, update, and close complaints efficiently.

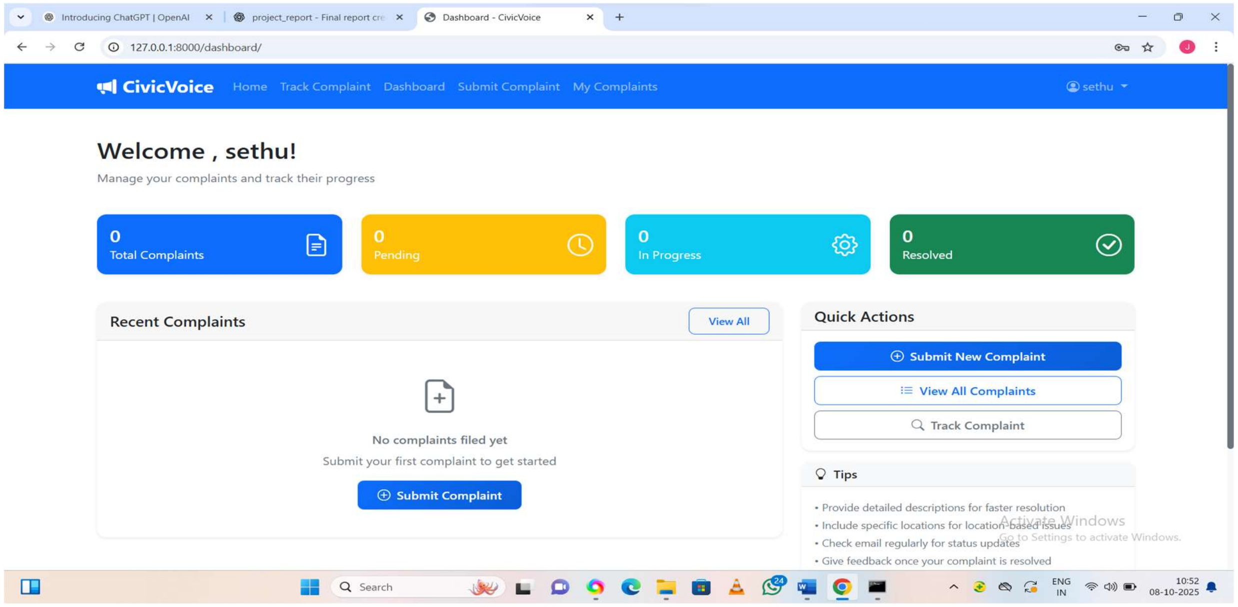


Make Your Voice Heard with CivicVoice

Transform how you interact with government services. File complaints, track progress in real-time, and drive meaningful change in your community through our transparent digital platform.

Figure 4.1: home page

Figure 4.1 shows the landing page of the **Civic Voice** application. This is a public web page where users can access options to register or log in to the system. It provides an introduction to the platform and allows users to track the status of complaints using their complaint ID.



Welcome , sethu!

Manage your complaints and track their progress

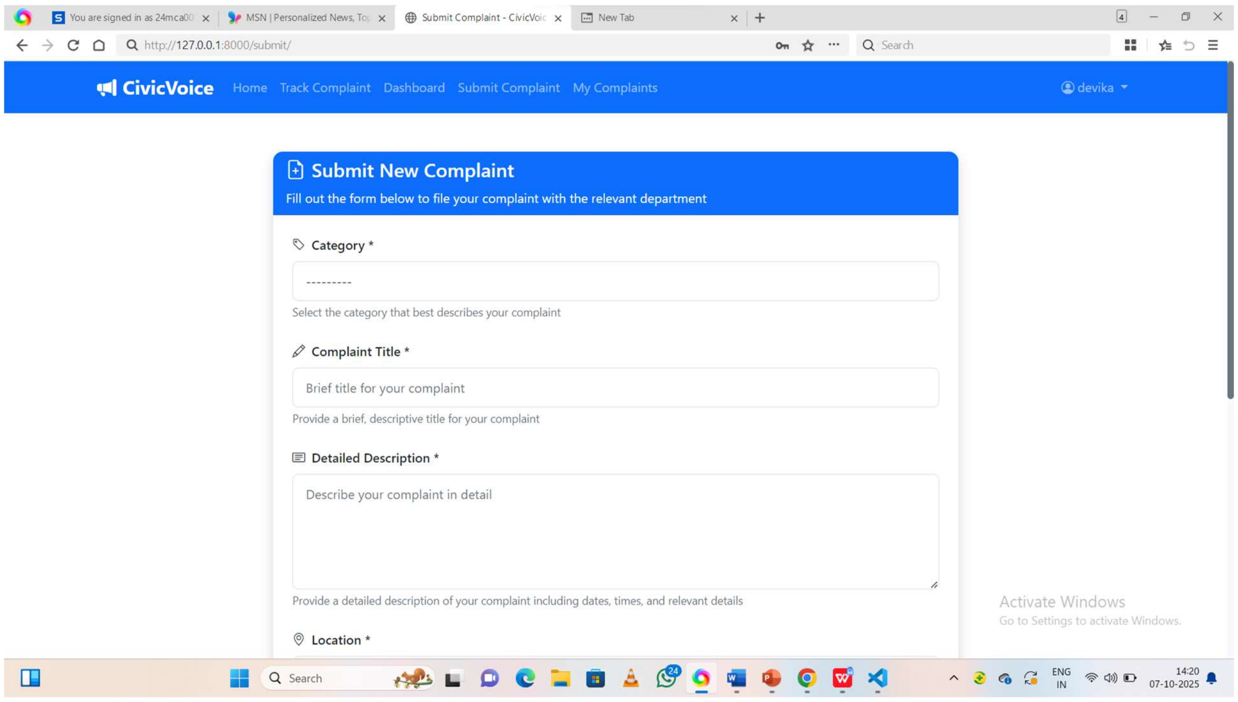
No complaints filed yet

Submit your first complaint to get started

- Provide detailed descriptions for faster resolution
- Include specific locations for location-based issues
- Check email regularly for status updates
- Give feedback once your complaint is resolved

Figure 4.2:user interface

Figure 4.2 shows the user dashboard of the Civic Voice application. It displays a summary of sthe user’s submitted complaints, including their total, pending, in-progress, and resolved counts. The interface allows users to view complaint details, submit new complaints, and track their progress efficiently.



Submit New Complaint
Fill out the form below to file your complaint with the relevant department

Category *

Select the category that best describes your complaint

Complaint Title *
Brief title for your complaint
Provide a brief, descriptive title for your complaint

Detailed Description *
Describe your complaint in detail
Provide a detailed description of your complaint including dates, times, and relevant details

Location *

Figure 4.3: compliant submission form

Figure 4.3 shows the Submit New Complaint page of the Civic Voice system. This page allows users to enter complaint details such as category, title, description, and location. It ensures that all required information is collected to route the complaint accurately to the concerned department for action

Chapter 5. Conclusion

The project Civic Voice was developed to provide a transparent, citizen-centric platform for managing public grievances through a centralized and efficient web-based system. The platform enables users to register complaints related to various government departments, track their progress in real-time, and receive timely updates on resolutions. Unlike traditional manual grievance handling processes, this system ensures accountability, data security, and accessibility while offering a user-friendly interface for both citizens and administrators. The system successfully integrates essential features such as user registration and login, complaint submission, automatic department routing, status tracking, and feedback collection. The administrative interface provides tools for authorized officials to review complaints, update statuses, add remarks, and generate analytical reports. These features work together to ensure a smooth flow of information and faster resolution of citizen issues.

The results demonstrate that the system effectively achieves its objectives of improving communication between citizens and government departments, reducing manual workload, and promoting transparency. Citizens benefit from a simple interface that allows them to file and follow complaints conveniently, while administrators gain access to organized dashboards and tracking mechanisms that improve their efficiency in managing cases.However, the system also presents certain limitations. Currently, it functions as a web-based application accessible through browsers, and it does not yet support a mobile application or real-time notification system. Integrating advanced features such as AI-based complaint prioritization, multilingual support, and mobile app accessibility in future versions can further enhance user convenience and government responsiveness .Overall, Civic Voice successfully demonstrates a digital approach to grievance management that aligns with the goals of e-governance and citizen empowerment.

References

[1] P. Sharma and A. Verma, "Design and Development of Online Grievance Redressal System for Public Sector," *International Journal of Computer Applications*, vol. 182, no. 25, pp. 10–15, 2021.

[2] R. Gupta and K. Singh, "E-Governance and Citizen Complaint Management Using Web-Based Systems," *International Journal of Advanced Research in Computer Science*, vol. 13, no. 2, pp. 45–50, 2022.

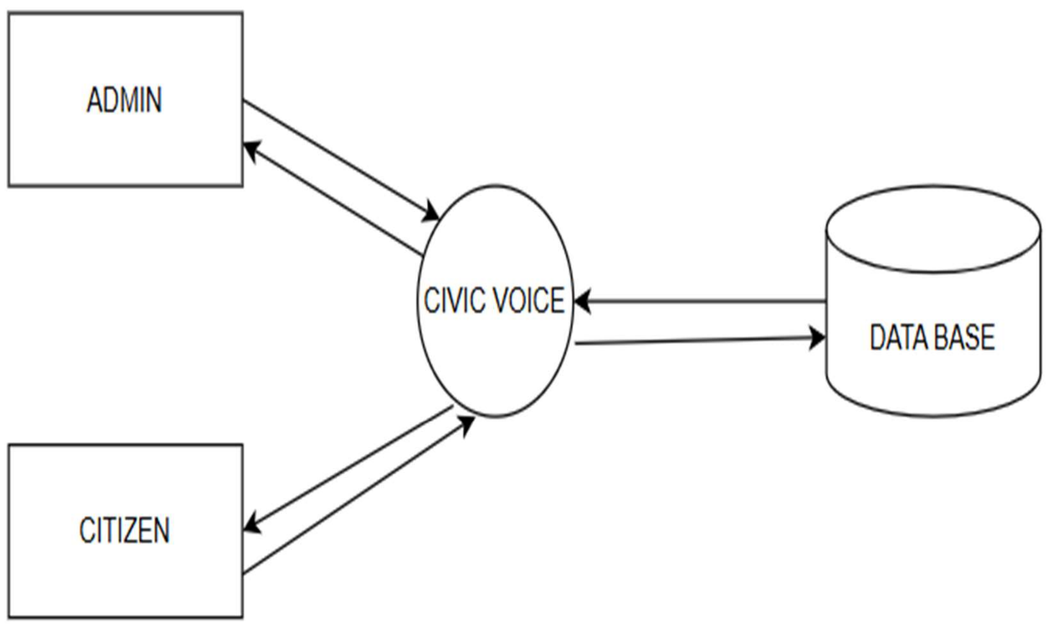
[3] "Django Official Documentation," *Django Software Foundation*, [Online]. Available: https://docs.djangoproject.com/. [Accessed 18 October 2024].

[4] "SQLite Documentation," *SQLite.org*, [Online]. Availablhttps://www.sqlite.org/docs.html. [Accessed 18 October 2024].

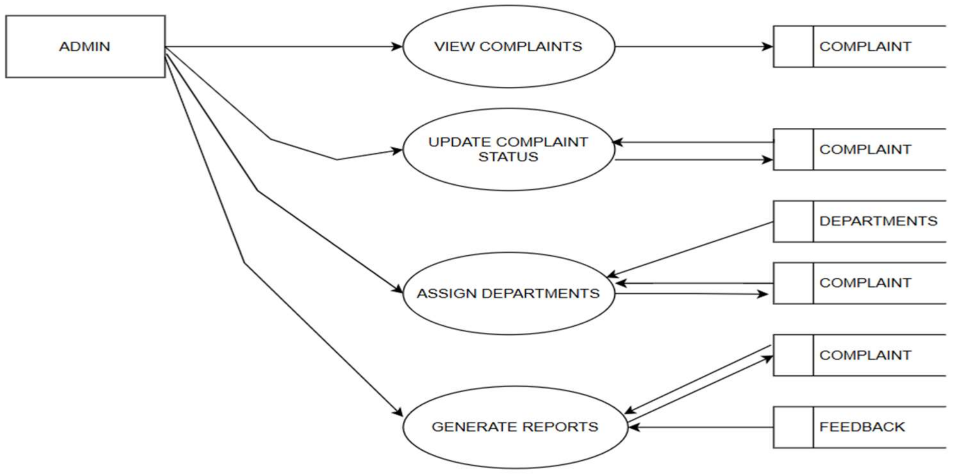
Appendix

Appendix A Data Flow Diagram

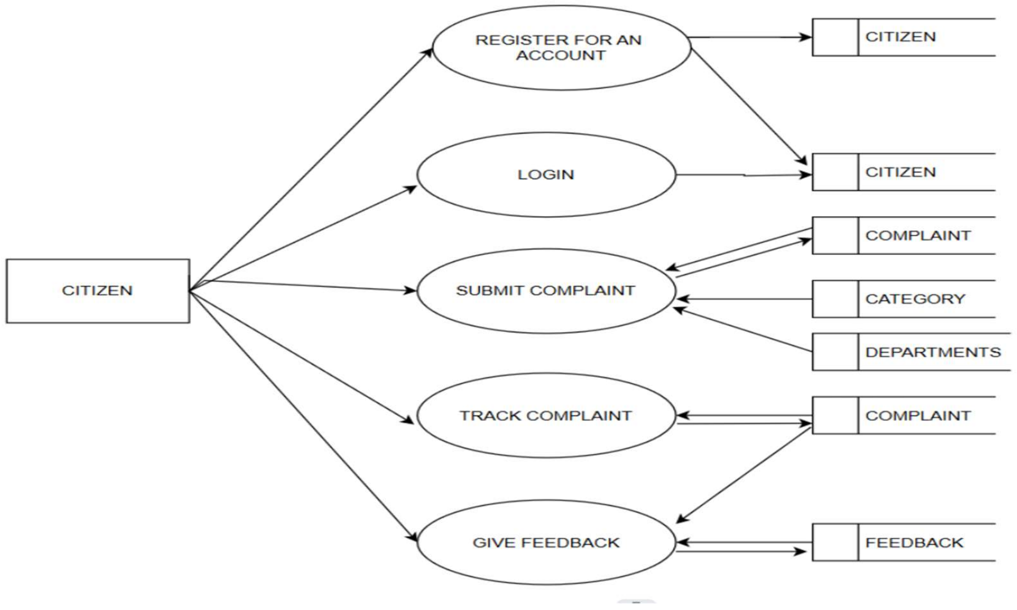
DFD Level 0



DFD Level 1

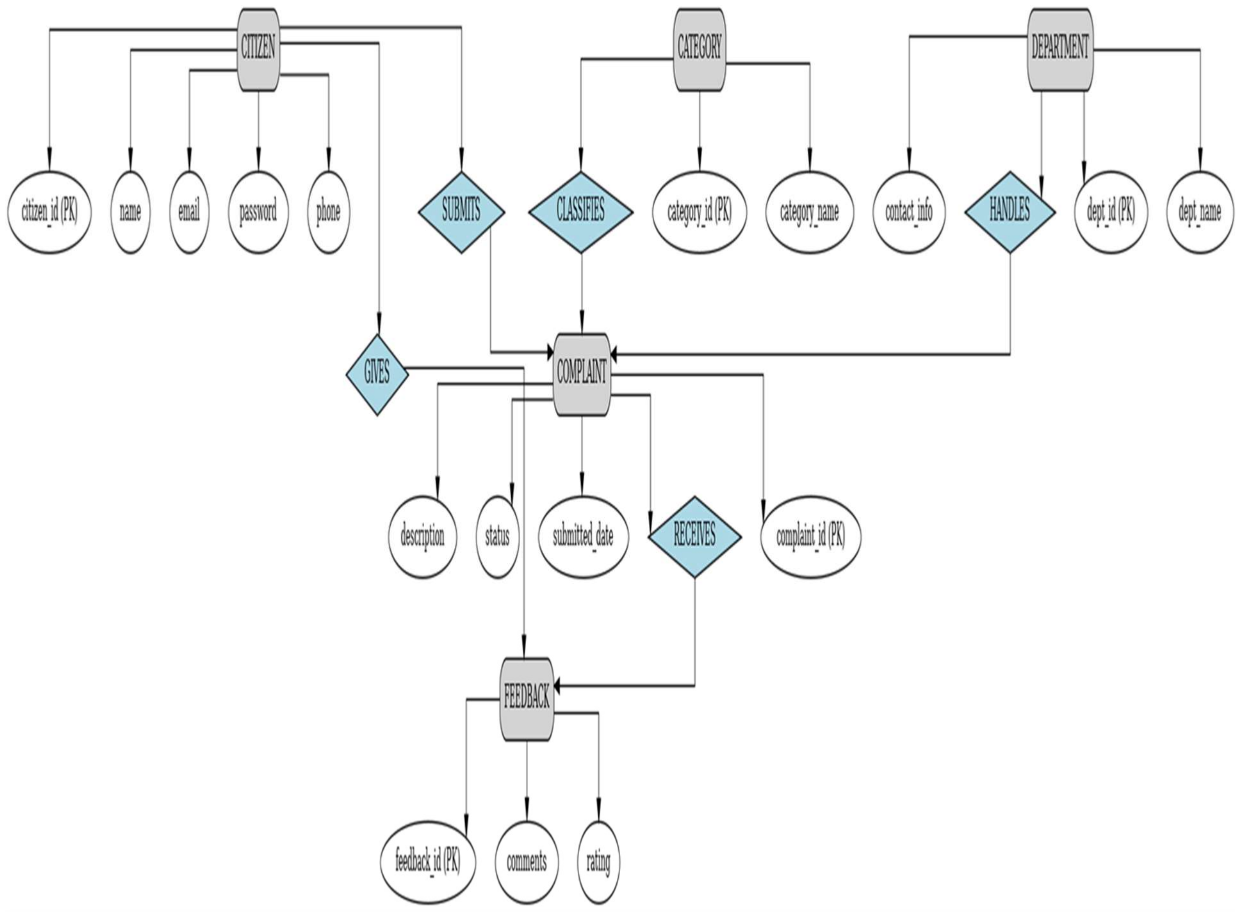


DFD Level 1.1



Appendix B

ER Diagram



Appendix C Source Code

Views.py

```
from django.shortcuts import render, redirect, get_object_or_404
from django.contrib.auth.decorators import login_required, user_passes_test
from django.contrib import messages
from django.core.mail import send_mail
from django.conf import settings
from django.db.models import Q
from django.http import JsonResponse
from .models import Complaint, Category, Feedback
from .forms import ComplaintForm, ComplaintSearchForm, FeedbackForm,
ComplaintStatusForm

def home(request):
    """Homepage view"""
    search_form = ComplaintSearchForm()
    return render(request, 'complaints/home.html', {'search_form':
search_form})

@login_required
def dashboard(request):
    """User dashboard showing their complaints and quick actions"""
    user_complaints = Complaint.objects.filter(user=request.user)[:5]  # Latest
5 complaints
    complaint_stats = {
        'total': Complaint.objects.filter(user=request.user).count(),
        'pending': Complaint.objects.filter(user=request.user,
status='PENDING').count(),
        'in_progress': Complaint.objects.filter(user=request.user,
status='IN_PROGRESS').count(),
        'resolved': Complaint.objects.filter(user=request.user,
status='RESOLVED').count(),
    }

    context = {
        'complaints': user_complaints,
        'stats': complaint_stats,
    }
    return render(request, 'complaints/dashboard.html', context)

@login_required
def submit_complaint(request):
    """Submit a new complaint"""
    if request.method == 'POST':
        form = ComplaintForm(request.POST)
        if form.is_valid():
            complaint = form.save(commit=False)
            complaint.user = request.user
            complaint.save()

            # Send email notification to user
            send_complaint_email(complaint)
            messages.success(request, f'Your complaint has been submitted
successfully! Complaint ID: {complaint.complaint_id}')
            return redirect('complaints:complaint_detail',
complaint_id=complaint.complaint_id)
        else:
            messages.error(request, 'Please correct the errors below.')
    else:
        form = ComplaintForm()

    return render(request, 'complaints/submit_complaint.html', {'form': form})

@login_required
def my_complaints(request):
    """Display all complaints by the logged-in user"""
    complaints = Complaint.objects.filter(user=request.user)
    return render(request, 'complaints/my_complaints.html', {'complaints':
complaints})

def track_complaint(request):
    """Track complaint by ID (public view)"""
    complaint = None
    if request.method == 'POST':
        form = ComplaintSearchForm(request.POST)
        if form.is_valid():
            complaint_id = form.cleaned_data['complaint_id']
            try:
                complaint = Complaint.objects.get(complaint_id=complaint_id)
            except Complaint.DoesNotExist:
                messages.error(request, f'No complaint found with ID:
{complaint_id}')
    else:
        form = ComplaintSearchForm()

    return render(request, 'complaints/track_complaint.html', {'form': form,
'complaint': complaint})

def complaint_detail(request, complaint_id):
    """Display complaint details"""
    complaint = get_object_or_404(Complaint, complaint_id=complaint_id)

    # Check if user can view this complaint (either owner or staff)
    if not (request.user.is_authenticated and (request.user == complaint.user
or request.user.is_staff)):
        messages.error(request, 'You do not have permission to view this
complaint.')
```

```
        return redirect('complaints:home')

    # Check if feedback exists
    feedback = None
    can_give_feedback = False

    if complaint.status == 'RESOLVED' and request.user == complaint.user:
        try:
            feedback = Feedback.objects.get(complaint=complaint)
        except Feedback.DoesNotExist:
            can_give_feedback = True

    context = {
        'complaint': complaint,
        'feedback': feedback,
        'can_give_feedback': can_give_feedback,
    }
    return render(request, 'complaints/complaint_detail.html', context)

@login_required
def give_feedback(request, complaint_id):
    """Give feedback for a resolved complaint"""
    complaint = get_object_or_404(Complaint, complaint_id=complaint_id,
user=request.user)

    if complaint.status != 'RESOLVED':
        messages.error(request, 'You can only give feedback for resolved
complaints.')
        return redirect('complaints:complaint_detail',
complaint_id=complaint_id)

    # Check if feedback already exists
    if Feedback.objects.filter(complaint=complaint).exists():
        messages.info(request, 'You have already provided feedback for this
complaint.')
        return redirect('complaints:complaint_detail',
complaint_id=complaint_id)

    if request.method == 'POST':
        form = FeedbackForm(request.POST)
        if form.is_valid():
            feedback = form.save(commit=False)
            feedback.complaint = complaint
            feedback.save()
            messages.success(request, 'Thank you for your feedback!')
            return redirect('complaints:complaint_detail',
complaint_id=complaint_id)
    else:
        form = FeedbackForm()

    return render(request, 'complaints/give_feedback.html', {'form': form,
'complaint': complaint})

# Admin/Staff views
@user_passes_test(lambda u: u.is_staff)
def admin_dashboard(request):
    """Admin dashboard for managing complaints"""
    complaints = Complaint.objects.all()[:20]  # Latest 20 complaints

    # Statistics
    stats = {
        'total': Complaint.objects.count(),
        'pending': Complaint.objects.filter(status='PENDING').count(),
        'in_progress': Complaint.objects.filter(status='IN_PROGRESS').count(),
        'resolved': Complaint.objects.filter(status='RESOLVED').count(),
    }

    # Filter by category if requested
    category_filter = request.GET.get('category')
    if category_filter:
        complaints = complaints.filter(category__id=category_filter)

    categories = Category.objects.all()

    context = {
        'complaints': complaints,
        'stats': stats,
        'categories': categories,
        'selected_category': category_filter,
    }
    return render(request, 'complaints/admin_dashboard.html', context)

@user_passes_test(lambda u: u.is_staff)
def admin_complaint_detail(request, complaint_id):
    """Admin view for complaint details with status update capability"""
    complaint = get_object_or_404(Complaint, complaint_id=complaint_id)

    if request.method == 'POST':
        form = ComplaintStatusForm(request.POST, instance=complaint)
        if form.is_valid():
            old_status = complaint.status
            complaint = form.save()

            # Send notification email if status changed
            if old_status != complaint.status:
                send_status_update_email(complaint, old_status)
                messages.success(request, f'Complaint status updated to
{complaint.get_status_display()}')
            else:
                messages.success(request, 'Complaint updated successfully')

            return redirect('complaints:admin_complaint_detail',
complaint_id=complaint_id)
    else:
        form = ComplaintStatusForm(instance=complaint)

    # Get feedback if exists
    feedback = None
    try:
        feedback = Feedback.objects.get(complaint=complaint)
    except Feedback.DoesNotExist:
        pass

    context = {
        'complaint': complaint,
        'form': form,
        'feedback': feedback,
    }
    return render(request, 'complaints/admin_complaint_detail.html', context)

# Email utility functions
def send_complaint_email(complaint):
    """Send email notification when complaint is submitted"""
    subject = f'Complaint Submitted - {complaint.complaint_id}'
    message = f'''
Dear {complaint.user.first_name},

Your complaint has been successfully submitted.

Complaint ID: {complaint.complaint_id}
Title: {complaint.title}
Category: {complaint.category.name}
Status: {complaint.get_status_display()}
```

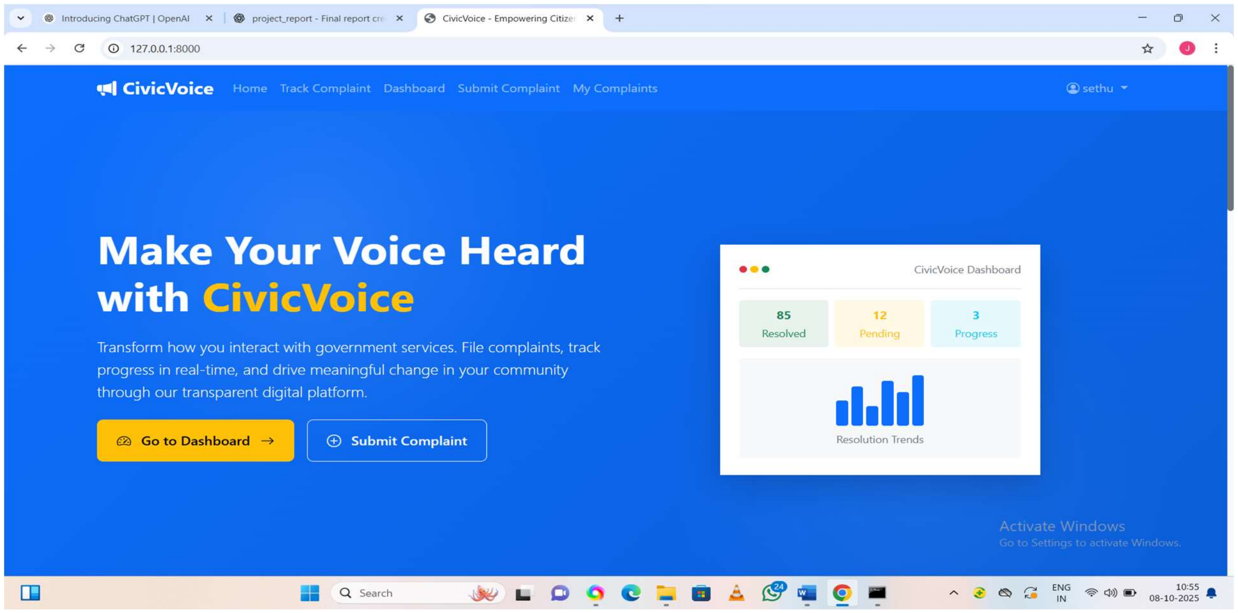
```
We will review your complaint and update you on the progress.

Thank you for using CivicVoice.

Best regards,
CivicVoice Team
    '''

    try:
        send_mail(
            subject,
            message,
            settings.DEFAULT_FROM_EMAIL,
            [complaint.user.email],
            fail_silently=False,
        )
    except Exception as e:
        print(f"Failed to send email: {e}")

def send_status_update_email(complaint, old_status):
    """Send email notification when complaint status is updated"""
    subject = f'Complaint Status Update - {complaint.complaint_id}'
    message = f'''
Dear {complaint.user.first_name},

Your complaint status has been updated.

Complaint ID: {complaint.complaint_id}
Title: {complaint.title}
Previous Status: {old_status.replace('_', ' ').title()}
New Status: {complaint.get_status_display()}

{f"Remarks: {complaint.admin_remarks}" if complaint.admin_remarks else ""}

Thank you for using CivicVoice.

Best regards,
CivicVoice Team
    '''

    try:
        send_mail(
            subject,
            message,
            settings.DEFAULT_FROM_EMAIL,
            [complaint.user.email],
            fail_silently=False,
        )
    except Exception as e:
        print(f"Failed to send email: {e}")
```

Models.py

```
from django.db import models
from django.contrib.auth.models import User
import uuid
from datetime import datetime

class Category(models.Model):
    name = models.CharField(max_length=100)
    description = models.TextField(blank=True)
    created_at = models.DateTimeField(auto_now_add=True)

    class Meta:
        verbose_name_plural = "Categories"

    def __str__(self):
        return self.name

class Complaint(models.Model):
    STATUS_CHOICES = [
        ('PENDING', 'Pending'),
        ('IN_PROGRESS', 'In Progress'),
        ('RESOLVED', 'Resolved'),
    ]

    user = models.ForeignKey(User, on_delete=models.CASCADE)
    category = models.ForeignKey(Category, on_delete=models.CASCADE)
    complaint_id = models.CharField(max_length=20, unique=True, editable=False)
    title = models.CharField(max_length=200)
    description = models.TextField()
    location = models.CharField(max_length=200)
    status = models.CharField(max_length=20, choices=STATUS_CHOICES,
default='PENDING')
    admin_remarks = models.TextField(blank=True, null=True)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    def save(self, *args, **kwargs):
        if not self.complaint_id:
            # Generate unique complaint ID
            year = datetime.now().year
            random_id = str(uuid.uuid4().hex[:6]).upper()
            self.complaint_id = f"CV{year}{random_id}"
        super().save(*args, **kwargs)

    class Meta:
        ordering = ['-created_at']

    def __str__(self):
        return f"{self.complaint_id} - {self.title}"

class Feedback(models.Model):
    RATING_CHOICES = [
        (1, 'Poor'),
        (2, 'Fair'),
        (3, 'Good'),
        (4, 'Very Good'),
        (5, 'Excellent'),
    ]

    complaint = models.OneToOneField(Complaint, on_delete=models.CASCADE)
    rating = models.IntegerField(choices=RATING_CHOICES)
    comments = models.TextField(blank=True)
    created_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f"Feedback for {self.complaint.complaint_id} -
{self.get_rating_display()}"
class UserProfile(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    phone_number = models.CharField(max_length=15, blank=True)
    address = models.TextField(blank=True)
    date_created = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f"{self.user.username} - Profile"

# Signal to create/update user profile
@receiver(post_save, sender=User)
def create_user_profile(sender, instance, created, **kwargs):
```

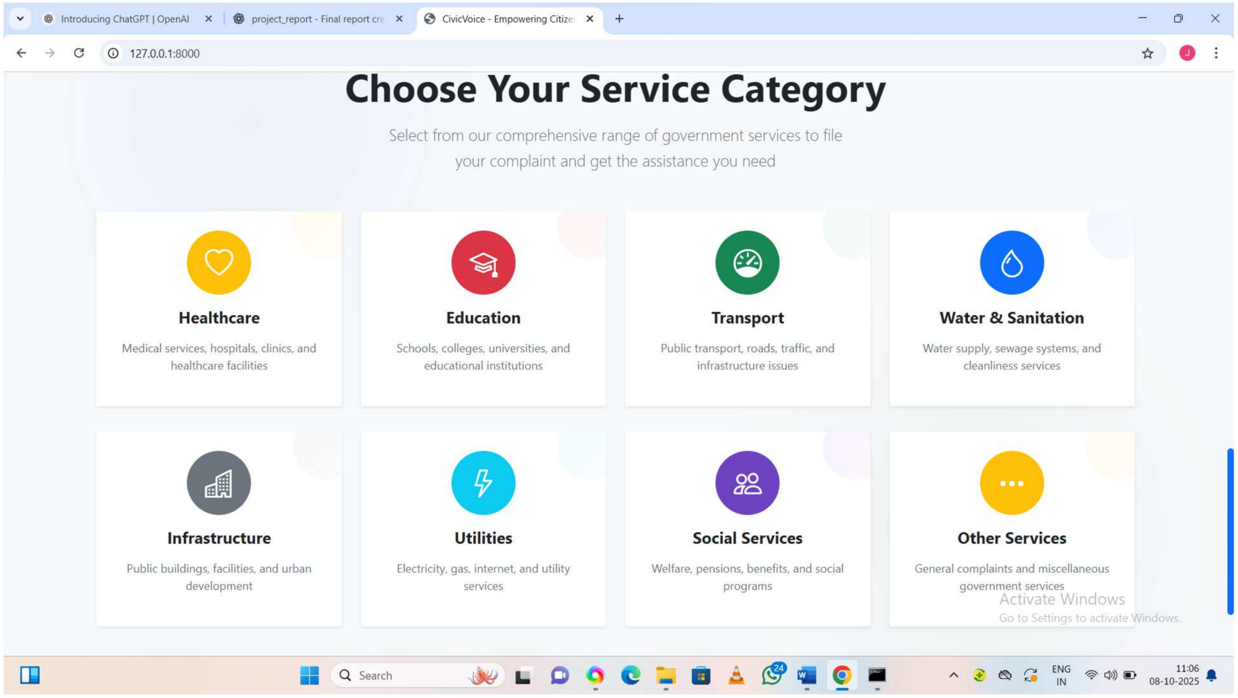
Appendix D. Screenshot

Hompage



Make Your Voice Heard with CivicVoice

Transform how you interact with government services. File complaints, track progress in real-time, and drive meaningful change in your community through our transparent digital platform.



Choose Your Service Category

Select from our comprehensive range of government services to file your complaint and get the assistance you need

Healthcare
Medical services, hospitals, clinics, and healthcare facilities

Education
Schools, colleges, universities, and educational institutions

Transport
Public transport, roads, traffic, and infrastructure issues

Water & Sanitation
Water supply, sewage systems, and cleanliness services

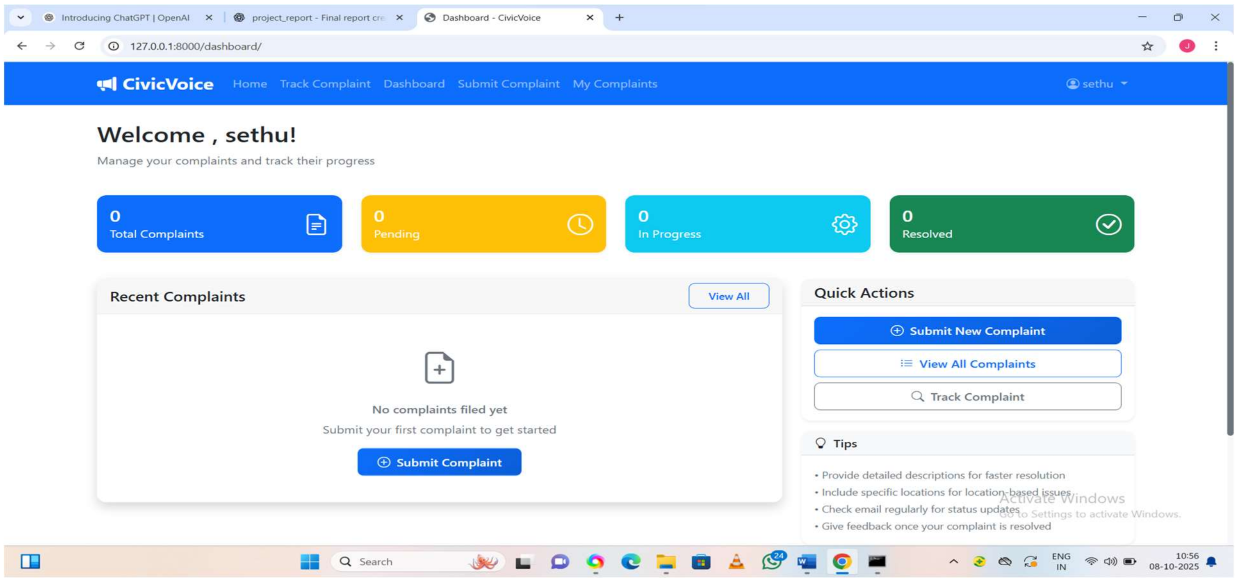
Infrastructure
Public buildings, facilities, and urban development

Utilities
Electricity, gas, internet, and utility services

Social Services
Welfare, pensions, benefits, and social programs

Other Services
General complaints and miscellaneous government services

User interface



Welcome , sethu!

Manage your complaints and track their progress

0 Total Complaints

0 Pending

0 In Progress

0 Resolved

Recent Complaints

No complaints filed yet

Submit your first complaint to get started

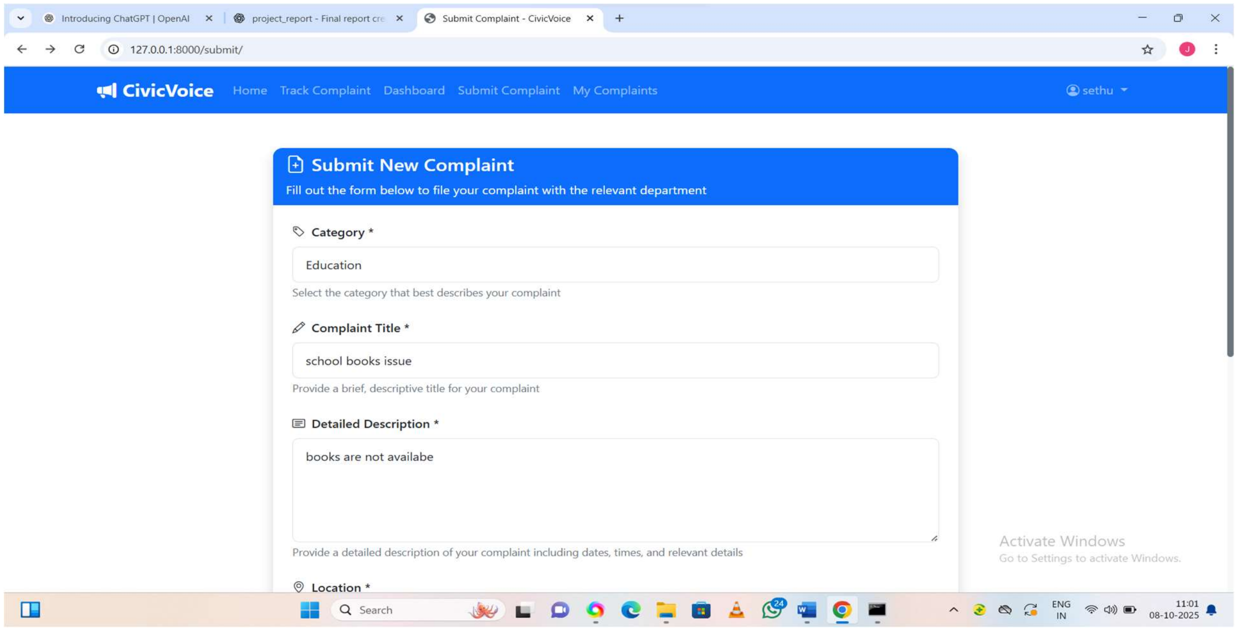
Quick Actions

Tips

- Provide detailed descriptions for faster resolution
- Include specific locations for location-based issues
- Check email regularly for status updates
- Give feedback once your complaint is resolved

Activate Windows
Go to Settings to activate Windows.

Complaint submission form



Submit New Complaint

Fill out the form below to file your complaint with the relevant department

Category *

Education

Select the category that best describes your complaint

Complaint Title *

school books issue

Provide a brief, descriptive title for your complaint

Detailed Description *

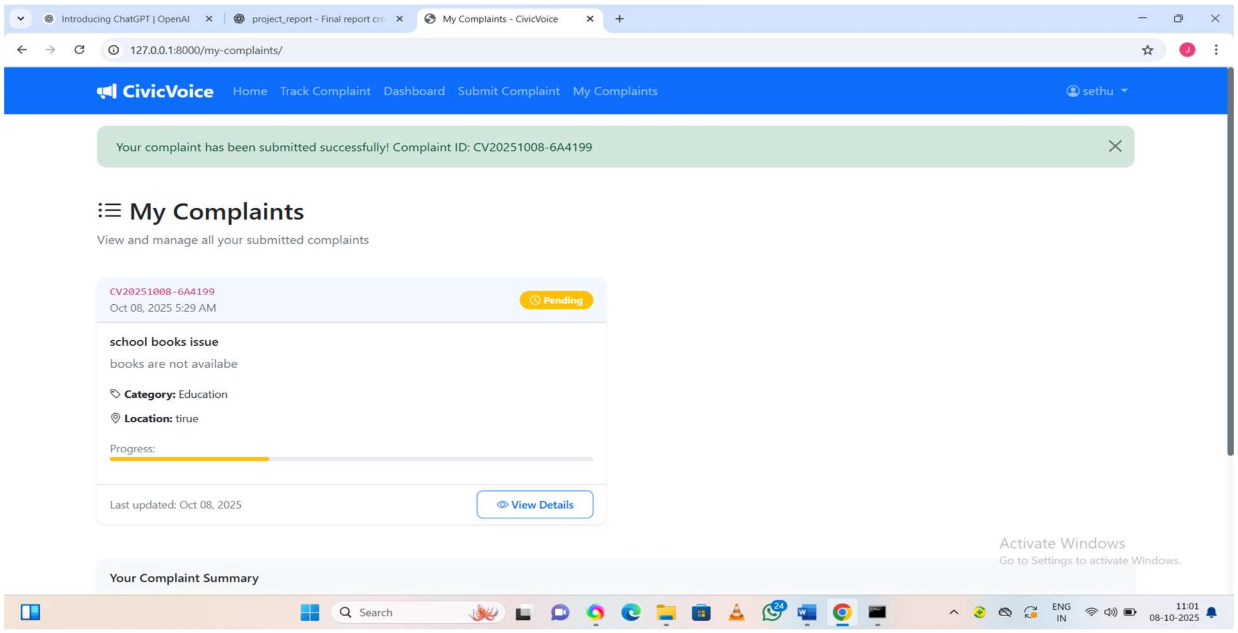
books are not availabe

Provide a detailed description of your complaint including dates, times, and relevant details

Location *

Activate Windows
Go to Settings to activate Windows.

Complaint ID generation


127.0.0.1:8000/my-complaints/

CivicVoice Home Track Complaint Dashboard Submit Complaint My Complaints sethu

Your complaint has been submitted successfully! Complaint ID: CV20251008-6A4199

My Complaints

View and manage all your submitted complaints

CV20251008-6A4199
Oct 08, 2025 5:29 AM
Pending

school books issue

books are not availabe

Category: Education

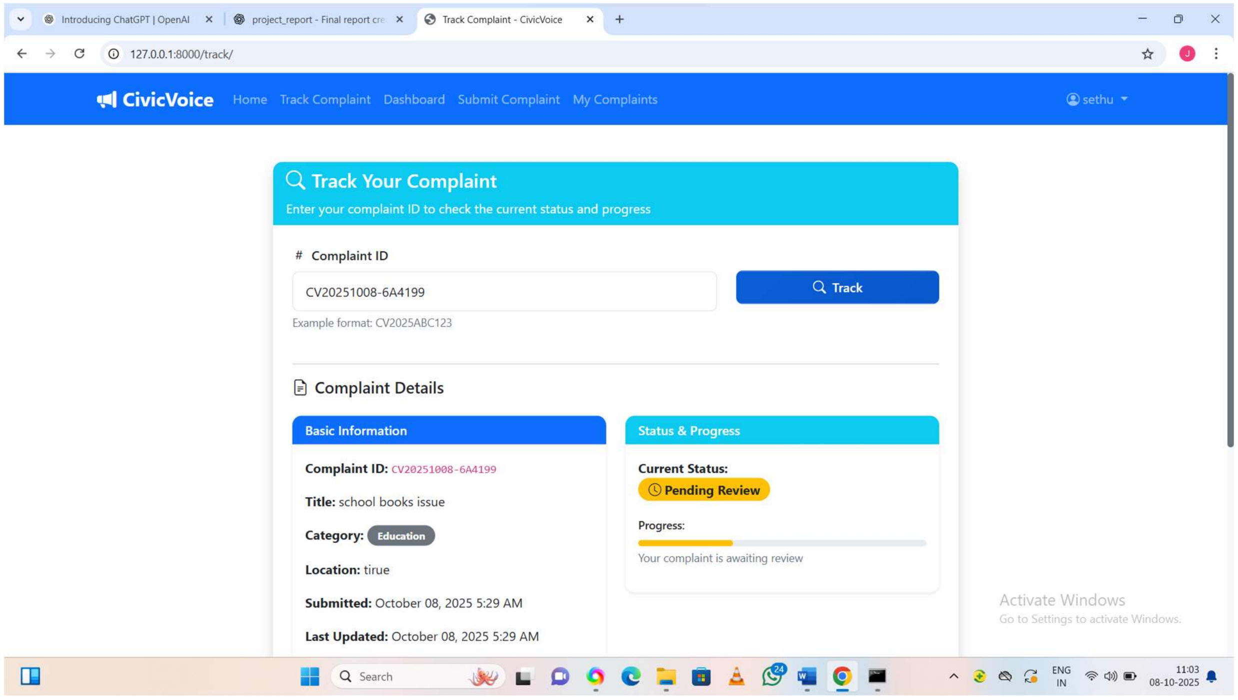
Location: tirue

Progress:

Last updated: Oct 08, 2025 View Details

Your Complaint Summary

Tracking complaint


127.0.0.1:8000/track/

CivicVoice Home Track Complaint Dashboard Submit Complaint My Complaints sethu

Track Your Complaint

Enter your complaint ID to check the current status and progress

Complaint ID: CV20251008-6A4199 Track

Example format: CV2025ABC123

Complaint Details

Basic Information

Complaint ID: CV20251008-6A4199

Title: school books issue

Category: Education

Location: tirue

Submitted: October 08, 2025 5:29 AM

Last Updated: October 08, 2025 5:29 AM

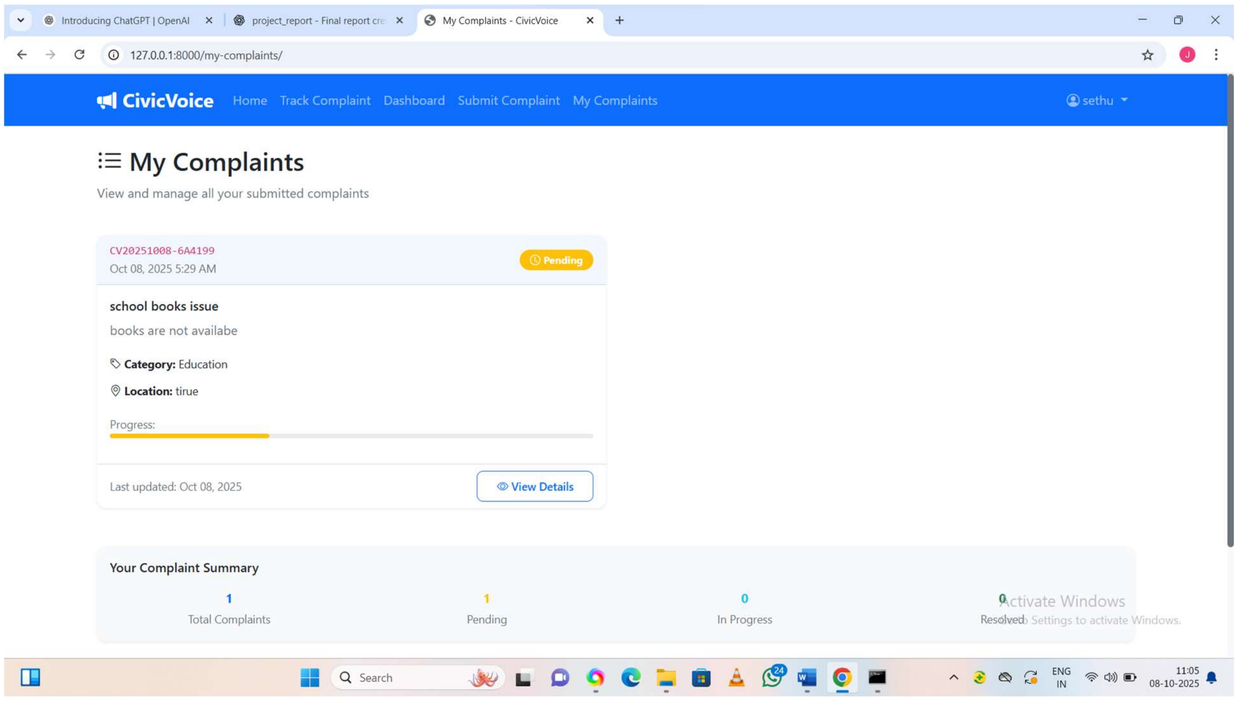
Status & Progress

Current Status: Pending Review

Progress:

Your complaint is awaiting review

Complaint history



My Complaints

View and manage all your submitted complaints

CV20251008-6A4199

Oct 08, 2025 5:29 AM

Pending

school books issue

books are not availabe

Category: Education

Location: tirue

Progress:

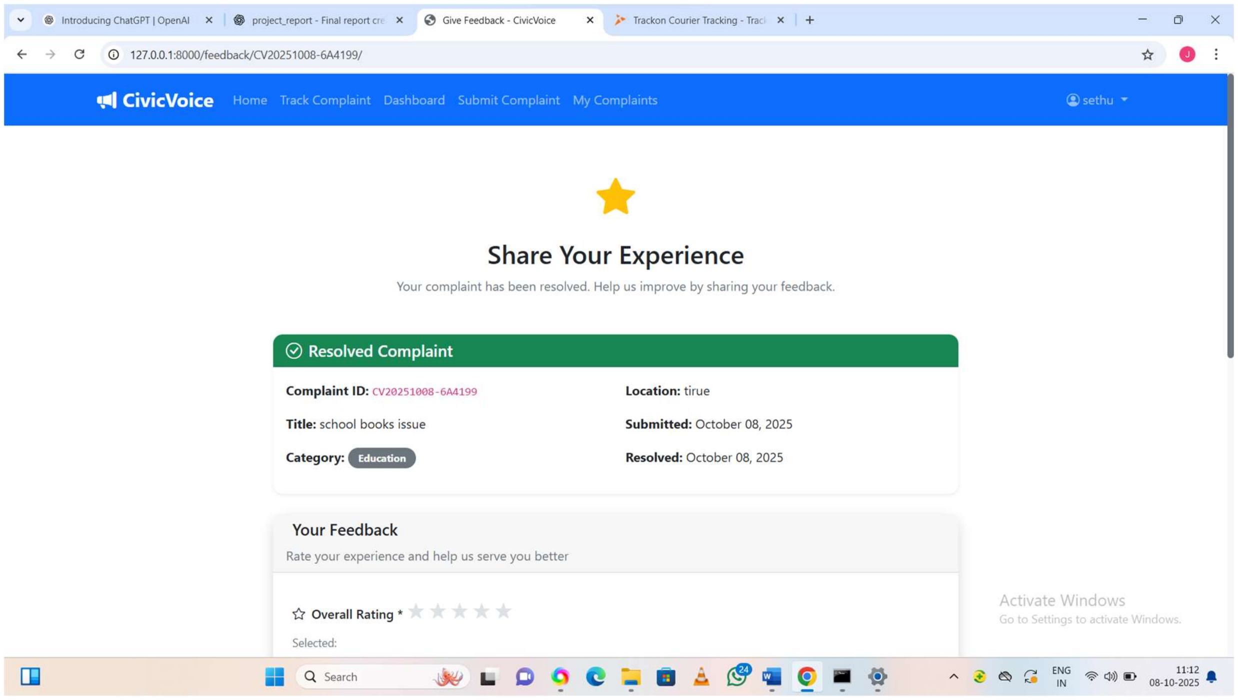
Last updated: Oct 08, 2025

View Details

Your Complaint Summary

1	1	0	0
Total Complaints	Pending	In Progress	Resolved

Feedback section



Share Your Experience

Your complaint has been resolved. Help us improve by sharing your feedback.

Resolved Complaint

Complaint ID: CV20251008-6A4199

Location: tirue

Title: school books issue

Submitted: October 08, 2025

Category: Education

Resolved: October 08, 2025

Your Feedback

Rate your experience and help us serve you better

Overall Rating *

Selected: