

Digital Complaint Management System (React + Django REST Framework)

1. Introduction

With the increasing shift toward digital services and e-governance, there is a growing need for transparent, secure, and efficient online systems that allow citizens to report issues and track their resolution. Traditional complaint handling systems are mostly manual, time-consuming, and lack accountability.

The **Digital Complaint Management System** is a web-based application developed using **React for the frontend** and **Python with Django REST Framework (DRF) for the backend**. The system enables users to submit complaints digitally and allows authorities (admins) to review, manage, and resolve them efficiently.

This project is designed to be **beginner to intermediate level**, while following **real-world industry architecture** where frontend and backend are separated.

2. Problem Statement

Many organizations and public offices still rely on manual methods for handling complaints, such as paper forms or verbal reporting. These approaches lead to:

- Delays in complaint resolution
- Lack of transparency
- Poor record management
- No proper tracking mechanism for users

A centralized digital platform is required to automate complaint registration, tracking, and resolution.

3. Objectives of the Project

The objectives of this project are:

- To provide an online platform for complaint submission
- To reduce manual paperwork and delays
- To ensure transparency in complaint handling
- To allow role-based access for users and administrators
- To help students understand full-stack development using React and DRF

4. Scope of the Project

The system can be applied in:

- Colleges and universities
- Local government offices
- Community-level issue reporting

- Small organizations

The project focuses on core functionalities and can be extended to a full e-governance system in the future.

5. System Architecture

The application follows a **client-server architecture**:

- **Frontend:** React.js (User Interface)
- **Backend:** Django REST Framework (API & Business Logic)
- **Database:** PostgreSQL (Primary data storage)

The frontend communicates with the backend using RESTful APIs over HTTP.

6. User Roles

6.1 User (Citizen / Student)

- Register and login
- Submit complaints
- View submitted complaints
- Track complaint status

6.2 Admin (Authority)

- Secure login
- View all complaints
- Update complaint status
- Add remarks or resolution notes

7. Functional Requirements

User Functionalities

- User registration and authentication using JWT
- Submit complaints with title, category, and description
- View complaint history
- Track status (Pending, In Progress, Resolved)

Admin Functionalities

- View all complaints
- Update complaint status
- Manage complaint records

8. Non-Functional Requirements

- Secure authentication and authorization

- Responsive user interface
- Reliable data storage
- Scalable backend architecture
- User-friendly design

9. Technology Stack and Tools

Frontend

- React.js
- JavaScript
- HTML and CSS (Tailwind CSS optional)
- Axios for API communication

Backend

- Python
- Django
- Django REST Framework
- JWT Authentication

Database

- PostgreSQL

10. Database Design

10.1 User Table

- id
- name
- email
- password
- role (user/admin)

10.2 Complaint Table

- id
- user_id (Foreign Key)
- title
- description
- category
- status
- created_at

11. Application Workflow

1. User registers and logs in

2. User submits a complaint
3. Complaint status is set to “Pending”
4. Admin reviews the complaint
5. Admin updates the status
6. User views updated status

12. Project Division (Team Work)

Member 1 – Backend Developer (Python + DRF)

- Database design
- API development
- Authentication and authorization
- Admin logic

Member 2 – Frontend Developer (React)

- UI design
- React components and routing
- API integration
- Dashboard implementation

13. Future Enhancements

- File upload for complaint evidence
- Email/SMS notifications
- Role-based permissions
- Analytics and reporting dashboard
- Cloud deployment

14. Advantages of the System

- Reduces paperwork
- Improves transparency
- Faster complaint resolution
- Secure and scalable
- Easy to use

15. Conclusion

The Digital Complaint Management System developed using React and Django REST Framework provides a practical solution for digital complaint handling. It demonstrates how modern web technologies can be used to build scalable, secure, and transparent systems. This project also serves as a strong foundation for advanced applications in e-governance and cloud-based systems.