GRIEVANCE PORTAL

A PROJECT REPORT

Submitted By:-

Rohan Kumar Mondal(23BCS10647)

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE

Chandigarh University
June,2025

BONAFIED CERTIFICATE

Certified that this project report of "**Bus Management system**" is the bonafied work of **Rohan Mondal(23BCS10647)** who carried out the project work under my/our supervision.

SIGNATURE

HOD NAME

HEAD OF DEPARTMENT

3rd year CSE

SIGNATURE

Er. Kamal Sharma

PROFESSOR

3rd year CSE

Submitted for the project viva-voice examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

TABLE OF CONTENT

LIST OF TABLES

ABSTRACT

The *Grievance Redressal Portal* is a full-stack web application designed to streamline the process of lodging, managing, and resolving public grievances within a government framework. Traditional complaint systems often rely on manual handling, resulting in delays, inefficiency, and lack of transparency. This project leverages modern web technologies to build a secure, user-friendly, and scalable system that allows citizens to register grievances and track their status while enabling administrators to monitor, prioritize, and resolve them efficiently. The portal integrates a **React + Vite** frontend for responsive, real-time interaction and a **Spring Boot (Java)** backend with an **H2 relational database** for persistent storage and scalable data management. The project demonstrates the implementation of a centralized complaint management solution aligning with the vision of Digital India, ensuring accountability, transparency, and citizen empowerment.

CHAPTER 1 : INTRODUCTION

1.1 Client Identification

The client for this project is the *Department of Administrative Reforms and Public Grievances (DARPG), Government of India* — a fictional representation of an organization responsible for improving governance and addressing citizen grievances. The portal is designed as an internal web-based system for handling complaints from citizens regarding public services and administration.

1.2 Identification of Problem

Manual grievance handling systems are inefficient, time-consuming, and prone to errors. Citizens face challenges such as lack of complaint tracking, inconsistent responses, and delayed redressal. Additionally, administrators lack tools to categorize, monitor, and analyze grievances effectively. This results in poor transparency and reduced citizen trust.

1.3 Identification of Tasks

Key tasks identified during system development include:

- Designing an intuitive user interface for grievance submission and tracking.
- Developing a secure backend to handle user authentication and grievance management.
- Integrating a database for structured complaint storage and retrieval.
- Implementing an administrative dashboard for monitoring and analytics.
- Ensuring role-based access for citizens, officers, and administrators.
- Validating user input and securing API endpoints using JWT authentication.

1.4 Organization of the Report

The report is divided into five chapters:

- Chapter 1 introduces the project and its objectives.
- Chapter 2 provides a literature review and existing research.
- Chapter 3 explains the design, methodology, and implementation details.
- Chapter 4 presents the result analysis and validation outcomes.
- Chapter 5 concludes with future scope and recommendations.

CHAPTER 2 : LITERATURE REVIEW/BACKGROUND STUDY

2.1 Timeline of the Reported Problem

Grievance management systems have evolved from manual registers to online web portals. However, existing portals often lack automation, analytical dashboards, and real-time updates. Over the last decade, the demand for e-Governance tools has driven innovation in citizen service delivery.

2.2 Proposed Solutions

Several government portals like CPGRAMS and state-level grievance systems have inspired this project. However, these systems often have outdated interfaces or limited transparency. The proposed *Grievance Redressal Portal* addresses these shortcomings with:

- A modern and responsive web interface.
- Real-time status tracking.
- Secure authentication and role-based access control.
- Administrative analytics to improve performance.

2.3 Bibliometric Analysis

Research papers and e-Governance reports highlight a growing emphasis on digital citizen services, transparency, and workflow automation. Studies reveal that integrating data visualization and machine learning can enhance grievance prioritization and reduce resolution time.

2.4 Review Summary

The review indicates that while many portals exist, few offer robust integration between frontend usability and backend efficiency. This project bridges that gap by providing a modular, scalable, and user-friendly solution.

2.5 Problem Definition

To design and implement a **centralized web-based platform** that allows citizens to lodge grievances, enables administrators to resolve and track them efficiently, and ensures data transparency and accountability in governance.

2.6 Goals / Objectives

- Enable citizens to register and track grievances online.
- Provide administrators with tools to manage and resolve complaints.
- Maintain a secure, scalable, and accessible platform.
- Implement status notifications and reporting dashboards.
- Simplify integration with future mobile or AI-based systems.

CHAPTER 3 : DESIGN FLOW AND PROCESS

3.1 Evaluation & Selection of Specifications/Features

Frontend:
React and Vite were chosen for fast development and modern tooling. Tailwind CSS ensures a consistent and responsive UI design. Axios handles API calls to the backend.

Backend:
Spring Boot provides an enterprise-level framework with integrated support for RESTful APIs, security (Spring Security + JWT), and database connectivity via JPA.

Database:
H2 was used for development due to its in-memory performance. The schema supports migration to MySQL for production environments.

3.2 Design Constraints

- Limited time for implementation within the semester schedule.
- Need for local development setup due to restricted cloud access.
- Security concerns around user data (addressed with JWT and validation).
- Academic infrastructure without cloud hosting (localhost environment).

3.3 Analysis and Feature Finalization Subject to Constraints

Finalized features include:

- User registration and login.
- Complaint submission with attachments.
- Real-time grievance tracking.
- Admin dashboard with status overview.
- Analytics chart for complaint types and resolution time.

3.4 Design Flow

System Architecture:

```
[Frontend: React + Vite]
        ↓
[REST API: Spring Boot Backend]
        ↓
[Database Layer: H2/MySQL]
        ↑
[Admin Dashboard & Reports]
```

The system follows the Model-View-Controller (MVC) architecture:

- Model → Entity classes (Grievance, User)
- View → React components
- Controller → Spring REST controllers

3.5 Design Selection

- **UI:** Tailwind CSS for clean layout.
- **Authentication:** JSON Web Tokens (JWT).
- **Routing:** React Router DOM.
- **Data Management:** Axios for API calls, JPA for ORM.

3.6 Implementation Plan / Methodology

Agile methodology was adopted with iterative development and testing cycles:

1. Requirement Analysis
2. System Design
3. Development (Frontend + Backend)
4. Integration Testing
5. Deployment & Feedback

CHAPTER 4 : RESULT ANALYSIS AND VALIDATION

4.1 Implementation of Solution

The final portal successfully integrates both frontend and backend components.

Frontend Output:

- Login/Signup pages built using React and Tailwind.
- Complaint submission form with validation.
- Status tracking and user dashboard.

Backend Output:

- REST endpoints for user registration, login, grievance submission, and status updates.
- Spring Security implemented for role-based access (Citizen/Admin).
- JWT used for secure token-based authentication.

Testing:

- Unit testing conducted for backend APIs using JUnit.
- Manual UI testing performed on multiple browsers.
- Validation ensured proper CRUD operations and error handling.

Performance:

- Average response time: <200ms for CRUD operations on local server.
- Database retrieval and insertions optimized via indexed columns.

Result Validation

During the validation phase, multiple test cases were executed to ensure that the *Grievance Redressal Portal* performed as expected under various user interactions.
The **user login** functionality was tested first, and upon entering valid credentials, the system successfully redirected the user to their respective dashboard, confirming proper authentication flow.
Next, the **grievance submission** process was verified by submitting multiple complaints through the frontend interface. Each submission was correctly stored in the database, validating the integrity of backend data handling and API communication.
Further testing of **admin updates** demonstrated that any status change or remark made by an administrator was immediately reflected on the citizen's dashboard, ensuring synchronization between frontend and backend layers.
Lastly, the **unauthorized access** test confirmed that users without proper authentication tokens were denied access to restricted routes, validating the effectiveness of the implemented JWT-based security mechanism.

Overall, all the planned test cases were executed successfully, and the portal passed each scenario, demonstrating robustness, accuracy, and reliability across its major functionalities.

CHAPTER 5 : CONCLUSION AND FUTURE WORK

5.1 Conclusion

The *Grievance Redressal Portal* fulfills its objective of providing a digital medium for grievance handling. It bridges the gap between citizens and administrators by ensuring transparency, accountability, and convenience. The system successfully demonstrates full-stack integration with modern tools and adheres to e-Governance principles. The academic implementation validates the feasibility and scalability of such systems for public administration.

5.2 Future Work

Future enhancements could include:

- Integration with **MySQL/PostgreSQL** for production-grade persistence.
- Addition of **AI-based complaint categorization** and prioritization.
- **Mobile application** (React Native) for accessibility.
- **Email/SMS notifications** for grievance status.
- Integration with **data analytics dashboards** using Power BI or Grafana.
- Cloud deployment via AWS or Azure with CI/CD pipelines