Student Grievance Portal

A PROJECT REPORT

Submitted by

Gagnesh Kakkar(23BCS11196)

in partial fulfilment for the award of the degree of

Bachelors of Engineering
IN
Computer Science Engineering

TABLE OF CONTENTS

OVERVIEW

Title of Project:

Student Grievance Portal

DOMAIN: Web Development, Student Services Management

Description:

The Student Grievance Portal is a comprehensive web-based application designed to streamline and modernize the grievance management process within educational institutions. In traditional systems, students often face difficulties such as slow response times, lack of transparency, and the absence of proper records when submitting complaints or feedback. These inefficiencies can create frustration and weaken trust between students and institutional authorities.

This portal provides a structured digital medium through which students can submit grievances or feedback anytime and from anywhere. Each grievance is recorded systematically, ensuring that no concern is overlooked or lost. The platform features a secure login system, enabling personalized access where students can log in to raise concerns, upload supporting details, and track the progress of their submissions in real time.

On the administrative side, the portal offers a centralized dashboard for faculty or designated administrators to review, manage, and respond to grievances efficiently. Complaints can be categorized, prioritized, assigned to responsible staff, and updated at each stage until fully resolved. This promotes transparency, accountability, and timely action—key factors in maintaining a healthy student-institution relationship.

Overall, the system enhances communication, promotes fairness, prevents delays, and builds an environment where student concerns are acknowledged and handled efficiently.

Technology:

Layer	Technology Used	Purpose
Frontend	React.js	Builds a dynamic, interactive, and responsive UI for students and admins
Backend	Spring Boot (Java)	Handles business logic, authentication, and API communication
Database	PostgreSQL	Stores users, grievances, messages, and status logs securely and efficiently

Layer	Technology Used	Purpose
Security	JWT (JSON Web Token) Authentication	Ensures secure and role-based access control for Students and Admin users

Objective:

The main objective of the Student Grievance Portal is to create an efficient, reliable, and transparent communication framework between students and the institution. By leveraging modern web technologies, the system ensures:

- **Quick submission** and acknowledgement of grievances
- **Clear tracking and visibility** into the status of each complaint
- **Accountability** from administrative authorities
- **A structured and organized resolution process**

Ultimately, the goal is to enhance student satisfaction, foster trust, and promote a supportive academic environment where student voices are valued and addressed responsibly.

CHAPTER 1.

INTRODUCTION

Grievance management is an essential component of student support and institutional governance within educational environments. Students may encounter issues related to academics, administration, infrastructure, peer interactions, or faculty conduct. When such concerns are not addressed in a timely and transparent manner, it can lead to dissatisfaction, reduced motivation, and loss of trust in the institution's administration. Ensuring that students feel heard and valued is therefore fundamental to maintaining a positive learning atmosphere.

However, traditional grievance handling methods—such as handwritten application submissions, manual complaint registers, or informal verbal reporting—tend to be slow and inefficient. These methods often lack transparency, as students are rarely informed about the status or progress of their complaints. Additionally, administrators may find it difficult to track and prioritize grievances effectively, which can lead to delays, miscommunication, or even loss of records. In the absence of a structured system, monitoring patterns or generating analytical insights for institutional improvement becomes nearly impossible.

The Student Grievance Portal addresses these challenges by digitalizing the entire grievance life cycle. The platform ensures that grievances are submitted systematically, stored securely, and processed in an organized manner. Students can submit complaints online from anywhere, view the real-time status of their submissions, and receive timely responses from the institution. Administrators, on the other hand, can review grievances through a centralized dashboard, categorize them based on priority, assign them to responsible staff members, and update progress openly.

By enabling transparency, traceability, and accountability, the portal strengthens communication between students and the administration. It promotes fair resolution practices, improves decision-making efficiency, and ultimately enhances student satisfaction and institutional credibility.

CHAPTER 2.

SYSTEM MODULES

The Student Grievance Portal is designed using a modular architecture to ensure clarity of operations, scalability, and ease of maintenance. Each module performs a specific role while interacting seamlessly with others to provide a smooth and transparent grievance-handling workflow. The major functional modules of the system are described below:

2.1 Authentication & User Management

This module is responsible for providing secure access to the system. It supports two primary user roles: Student and Admin. Students can submit and track grievances, while Admins are authorized to review, respond, and resolve complaints. The system uses JWT (JSON Web Token) based authentication, ensuring that user sessions are secure and protected from unauthorized access**.**

Key Functions:

- User registration and login
- Password encryption using BCrypt for protection of credentials
- Role-based access control to prevent unauthorized actions
- Session management using JWT tokens

This module ensures that only verified users can interact with the portal and that their actions are restricted based on assigned privileges.

2.2 Grievance Submission & Tracking

This module allows students to submit grievances through a structured online form, specifying the subject and description of the issue. Each grievance is recorded with a timestamp and assigned an initial status of NEW. Students can view the list of grievances they have submitted at any time and check their current resolution status.

Key Functions:

- Grievance submission form
- View submitted grievances
- Status tracking (NEW → IN_PROGRESS → RESOLVED)
- Timestamped record maintenance

This module ensures transparency and reduces uncertainty by keeping students informed about the progress of their issues.

2.3 Admin Dashboard & Status Management

The Admin Dashboard provides authorized personnel with tools to manage and resolve student grievances effectively. Administrators can view all grievances in a sorted or filtered format (e.g., pending, in-progress, resolved). They can update the status of each grievance and optionally assign responsibility to specific staff members for resolution.

Key Functions:

- View all grievances in tabular / dashboard layout
- Filter grievances by category or status
- Update grievance status at different resolution stages
- Assign grievances to relevant administrative staff (optional extension)

This module promotes accountability and structured issue resolution.

2.4 Feedback & Response Threading

This module facilitates two-way communication between the student and the admin regarding any specific grievance. Instead of a one-time response, it supports threaded messaging, enabling continuous conversation until the issue is resolved. Each message is stored with sender identity and timestamp to maintain transparency and clarity.

Key Functions:

- Students and admins can exchange responses on grievances
- Every message is linked to a grievance record
- Threaded UI for easy conversation tracking

This ensures clarity of communication and helps maintain proper discussion records.

2.5 Notification & Reports

The Notification & Reporting module enhances the usability and administrative insights of the system. Students receive alerts when their grievance status is updated, and administrators can generate summarized reports to understand grievance patterns and institutional concerns.

Key Functions:

- Status update notifications (e.g., email, alert pop-ups)

- Admin dashboard displaying total grievances by status/category
- Option to generate reports for decision-making or audits

This module supports data-driven analysis and ensures timely communication.

CHAPTER 3.

SYSTEM ARCHITECTURE & DESIGN

The Student Grievance Portal is designed using a **layered architecture** to ensure scalability, maintainability, and separation of concerns. Each layer is responsible for handling specific operations within the application, which improves performance and simplifies future enhancements. The system follows a three-tier architecture consisting of the **Frontend Layer**, **Backend Layer**, and **Database Layer**, which communicate with each other using standardized web protocols and structured data formats.

3.1 Frontend Layer (User Interface)

The frontend layer is developed using **React.js**, which provides an interactive and responsive user interface. It enables students and administrators to access the system via a web browser without requiring any local installations. React's component-based architecture allows the interface to be modular, reusable, and easy to maintain.
Key Responsibilities:

- Displaying forms for user login and grievance submission
- Providing dashboards for students and administrators
- Rendering real-time grievance status updates
- Managing navigation and user sessions using stored JWT tokens

This layer interacts only with the backend through API calls and does not directly access the database, ensuring security and proper access control.

3.2 Backend Layer (Application Logic & APIs)

The backend is implemented using **Spring Boot (Java)**, which provides a robust and scalable framework for building RESTful web services. This layer acts as the **application controller**, processing user inputs, enforcing business rules, and coordinating communication between the interface and the database.
Key Responsibilities:

- User authentication and authorization using JWT
- Handling grievance creation, retrieval, updates, and status transitions
- Managing communication threads for grievance responses
- Implementing role-based access (Student / Admin)
- Ensuring secure and reliable data transfer with HTTP REST APIs

The backend ensures **data validation, error handling, and data integrity** to maintain a consistent system state.

3.3 Database Layer (Data Storage and Persistence)

The database layer uses **PostgreSQL**, a secure and high-performance relational database system. All application data—including user accounts, grievance records, communication messages, timestamps, and status logs—are stored in structured tables.
Key Responsibilities:

- Persistent storage of user profiles and role information
- Storing and indexing all grievance records for efficient retrieval
- Maintaining threaded message histories for each grievance
- Ensuring data consistency and referential integrity through relational constraints

This layer interacts only with the backend and does not communicate directly with the frontend, preventing unauthorized data manipulation.

3.4 Overall System Workflow

1. A user (student/admin) logs into the system through the frontend.
2. The frontend sends login credentials to the backend.
3. The backend validates the credentials, generates a JWT token, and returns it to the frontend.
4. The student submits a grievance through the UI; the backend stores it in the database.
5. Admins access the dashboard through React UI to view and manage grievances.
6. Any updates made by the admin are stored in the database and reflected on the student dashboard.
7. Notification or status update is displayed to the student.

This architecture ensures **smooth communication, data security, and ease of access**, resulting in an efficient and transparent workflow.

CHAPTER 4.

IMPLEMENTATION & ANALYSIS

This chapter explains the implementation details of both the frontend and backend components of the Student Grievance Portal, along with the workflow and security measures adopted. The system has been developed using modern web technologies, ensuring efficiency, scalability, and maintainability.

4.1 Backend Implementation (Spring Boot)

The backend of the system is implemented using **Spring Boot**, which provides a robust framework for building RESTful APIs. The backend is responsible for handling all business logic, processing client requests, interacting with the database, and enforcing role-based access control.

4.1.1 Authentication and Security

To ensure secure access, the system uses **JWT (JSON Web Token) Authentication**, which protects API endpoints and restricts access based on user roles (Student or Admin).
Key Security Features:

- Passwords are encrypted using **BCrypt hashing**
- JWT tokens are generated upon successful login
- Tokens are verified with every request to authorize user privileges
- Students and Admins access different functionalities based on role-based authorization

This approach prevents unauthorized access and ensures secure communication between frontend and backend components.

4.1.2 REST API Endpoints

The backend provides well-structured REST APIs for all major operations, such as:

Operation	Endpoint	Method	Accessible By
User Login / Registration	/auth/login, /auth/register	POST	Student & Admin
Submit Grievance	/grievances	POST	Student
View Own Grievances	/grievances/student/{id}	GET	Student
View All Grievances	/grievances	GET	Admin
Update Grievance Status	/grievances/{id}/status	PUT	Admin

Operation	Endpoint	Method	Accessible By
Message Threading	/grievances/{id}/messages	POST	Student & Admin

The API layer ensures smooth interaction between frontend and backend while maintaining strict access control.

4.2 Frontend Implementation (React.js)

The frontend is developed using **React.js**, which enables the creation of an interactive and responsive interface. React's component-based architecture helps in reusability, modular design, and efficient rendering of UI elements.

4.2.1 Key Interface Components

- **Login & Registration Forms** – Enables authenticated user access
- **Student Dashboard** – Allows students to submit new grievances and track their status
- **Admin Dashboard** – Displays grievance lists with filtering options
- **Conversation Thread View** – Enables two-way messaging between student and admin

4.2.2 State Management

React Hooks and Context API are used to handle:
- Token-based session handling
- Dynamic rendering of grievance lists
- Communication thread updates

This results in a smooth, real-time user experience.

4.3 Database Design (PostgreSQL)

PostgreSQL is used for persistent and reliable data storage. The database contains structured tables with relational integrity.

Core Database Tables
- **User** *(id, username, password, role)*
- **Grievance** *(id, subject, description, status, student_id, submittedAt)*
- **Message** *(id, grievance_id, sender_id, content, timestamp)*

This structure ensures:
- Logical linkage between users and grievances
- Clear tracking of status changes
- Historical record of conversations for transparency

4.4 System Workflow Analysis

1. User logs in and receives JWT token.
2. Student submits grievance with description.
3. Backend stores grievance and marks status as **NEW**.
4. Admin reviews grievances through dashboard and updates status accordingly.
5. Both parties communicate via threaded messages until final resolution.
6. Status is updated to **RESOLVED**, completing the grievance cycle.

This workflow ensures organized and trackable grievance handling.

4.5 Performance and Reliability

- Backend APIs respond efficiently due to lightweight REST processing.
- Database indexing ensures quick retrieval of grievance records.
- JWT verification provides secure transaction handling.
- React UI updates content dynamically without full-page reloads.

The system achieves **high usability, transparency, and performance** across all modules.

CHAPTER 5.

CONCLUSION AND FUTURE WORK

5.1 Conclusion

The Student Grievance Portal provides an efficient, transparent, and systematic solution for managing student grievances within educational institutions. By digitalizing the entire grievance submission and response process, the system ensures that concerns raised by students are handled fairly and promptly. The platform eliminates the limitations of traditional paper-based and manual complaint-handling methods, such as delays, lack of traceability, and poor record maintenance.

The portal's secure authentication mechanism, intuitive user interface, structured grievance workflow, and searchable records enable both students and administrators to interact seamlessly. Students are empowered to submit grievances from anywhere and track the progress of their issues in real time. Administrators, on the other hand, benefit from a centralized management panel where grievances can be reviewed, categorized, assigned, and resolved efficiently.

Overall, the system enhances communication, promotes accountability, builds trust, and contributes to a more supportive academic environment. It stands as a valuable tool for maintaining institutional transparency and strengthening student satisfaction.

5.2 Future Work

While the current implementation fulfills the key requirements of grievance handling, there are several possibilities for improvement and feature enhancement. Future development directions include:

1. **AI-Based Grievance Prioritization**
   Implementing machine learning or natural language processing (NLP) techniques to analyze grievance content and automatically categorize or prioritize issues. This would help the administration identify urgent or recurring problems more effectively.

2. **Mobile Application Support**
   Developing a mobile version of the portal (Android/iOS) to increase accessibility and convenience for students and staff. A mobile app would enable real-time submissions and faster monitoring of grievance updates.

3. **Real-Time Notifications and Alerts**
   Integrating real-time notifications through web push services, SMS, or email. This ensures that students are immediately notified when the status of their grievance changes, and administrators receive alerts when new complaints are submitted.

4. **Report Generation and Analytics Dashboard (Extended)**
   Expanding the reporting module to include graphical analytics, such as trends,

recurring issue categories, and resolution time statistics. These insights can support policy improvements and strategic decision-making.

5. **Multi-Language and Accessibility Support**
   Adding language localization and accessibility features to make the platform usable for a wider and more diverse student audience.

This combination of completed outcomes and future development potential ensures that the Student Grievance Portal can continue to evolve into a more intelligent, scalable, and user-centric solution over time.