

Project Plan Document

Students Grievance Readdressal System (CRUD + REST APIs)

TEAM Number: 7

Members:

D N Vaishnavi – 22071A0572 -9121391366 - dnvaishnavi5@gmail.com

K Devi SriVarshini –22071A0573 - 7075767062 - varshinikarri9@gmail.com

K Anusha - 22071A0588 - 9676198317 - anush.varma05@gmail.com

M Nithya Sri Reddy - 22071A0594 - 7702827129 - nithyasrireddy129@gmail.com

P Anouksha Reddy - 22071A05A2-9494519234- anouksha.reddy@gmail.com

P Mohitha- 22071A05A4-7386560090- pusalamohitha@gmail.com

1. Project Overview

The Students Grievance Redressal System is a centralized web-based platform designed to help students effectively raise and track grievances related to academics, administration, or infrastructure. Students can submit categorized complaints, attach evidence, and monitor status updates. Administrators and higher authorities can view complaints daily, resolve them, forward to concerned departments, or escalate unresolved/serious issues. The system ensures structured communication, transparency, faster resolutions, and complete records for future review.

2. Objectives

- Provide secure student accounts for filing and tracking grievances.
- Enable complaint categorization, evidence attachment, and status tracking.

- Build an admin dashboard for complaint review, forwarding, and resolution.
 - Automate escalation for unresolved or serious complaints.
 - Provide analytics and reports (resolution rate, unresolved cases, complaint statistics).
 - Ensure transparency and accountability between students and authorities.
-

3. Timeline & Weekly Breakdown

Week 1: Requirement Analysis & Planning

- Define user roles: Student, Admin, Department Head, Higher Authority.
 - Finalize tech stack (Spring Boot + MySQL for backend, HTML/CSS/JavaScript for frontend)
 - Draft SRS (System Requirement Specification), flow diagrams, ER model.
 - Define modules: User Management, Complaint Handling, Admin Dashboard, Reports.
Deliverable: SRS Document, Flowchart, ER Diagram.
-

Week 2: Project Setup

- Initialize GitHub repository and branch strategy.
 - Set up Node.js/Express backend and MongoDB connection.
 - Set up React frontend with routing and authentication boilerplate.
 - Create project folder structure (backend & frontend separation).
Deliverable: Project skeleton with DB schema plan, GitHub repo setup.
-

Week 3: User Account Management

- Implement student registration & login (JWT authentication + bcrypt).
 - Create Student Dashboard to view personal complaint history.
 - Define models: User (student/admin/authority), Complaint.
Deliverable: Functional user login/registration with complaint history view.
-

Week 4: Complaint Submission Module

- Implement form to file new complaints (category + description + file upload).
 - Store complaint in MongoDB with timestamps & status.
 - API endpoints: POST /complaints, GET /complaints/:id.
Deliverable: Complaint submission form & database integration.
-

Week 5: Admin Dashboard & Complaint Categorization

- Admin can view daily complaint reports & details.
 - Option to mark complaints as *Resolved* or *Forward to Department*.
 - Implement forwarding logic to concerned departments.
Deliverable: Admin dashboard with categorization & forwarding functionality.
-

Week 6: Escalation & Notifications

- Implement automatic escalation for unresolved complaints > 7 days.
 - Notify higher authority for serious complaints.
 - Add email/SMS notification integration for status updates.
Deliverable: Escalation logic, notification system.
-

Week 7: Analytics & Reporting

- Display complaint statistics: Resolved, Pending, Escalated.
 - Graphical reports (pie charts, bar graphs) for admin.
 - Generate downloadable reports (PDF/CSV).
Deliverable: Functional analytics dashboard with exportable reports.
-

Week 8: Testing, Documentation & Deployment

- Conduct unit & integration testing.
 - Prepare user manual and API documentation (Swagger/Postman).
 - Optimize code & finalize database schema.
 - Deploy system using Render/Heroku (backend) & Netlify/Vercel (frontend).
Deliverable: Live demo URL, GitHub repo, documentation, final presentation.
-

4. Deliverables Checklist

Item	Description	Owner	Week Due
SRS & ER Diagram	System requirement specification & ER model	Team Lead	1
Project Setup	Folder structure, DB connection, repo	Dev 1	2
User Management	Login, registration, dashboard	Dev 2	3
Complaint Submission	File complaint module	Dev 3	4
Admin Dashboard	Categorization, forwarding system	Dev 1	5
Escalation & Notifications	Auto escalation, email/SMS alerts	Dev 2	6
Analytics & Reporting	Charts, reports export	Dev 3	7

5. Tools & Technologies

- **Frontend:** HTML, CSS, JavaScript (with Chart.js for analytics/graphs)
- **Backend Framework:** Spring Boot (Java)
- **Database:** MySQL (with JPA/Hibernate for ORM)
- **Authentication:** Spring Security + JWT
- **Charts & Analytics:** Chart.js (JavaScript library)
- **Testing:** Postman (API testing), JUnit (backend testing)
- **Documentation:** Swagger/OpenAPI, README
- **Deployment:** Apache Tomcat / Spring Boot embedded server (backend), hosted on AWS/Heroku; frontend can be hosted on GitHub Pages/Netlify
- **Notifications:** JavaMail API / Twilio (optional)