VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI, KARNATAKA -590 018

A Minor Project Report on

"COMPLAINT MANAGEMENT SYSTEM"

Submitted in partial fulfillment of the requirements for

5th Semester Minor Project

2025–26.

By

Afifa taskeen -4MH23CA002

Charitha h k -4MH23CA010

Under the Guidance of

MS SHANA

AST professor

Dept. of CSE (AI), MIT Mysore.

MIT Mysore.

2025-26DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE)MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE

An Autonomous Institute Affiliated to VTU Belagavi

MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE

DEPARTMENT OF CSE (Artificial Intelligence)

CERTIFICATE

Certified that the Minor Project work entitled “complaint management system”, was carried out by Mr./Ms. [charitha h k,Afifa taskeen], USN _4MH23CA010,4MH23CA002_____, a bonafide student of Maharaja Institute of Technology Mysore, in partial fulfillment of the requirements for the 5th Semester Minor Project for the academic year 2025–26.

It is certified that all corrections and suggestions indicated have been incorporated in the report. This Minor Project report meets the academic requirements prescribed for the said semester.

Signature of the Guide
Name
Designation
Dept. of CSE (AI)

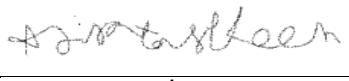
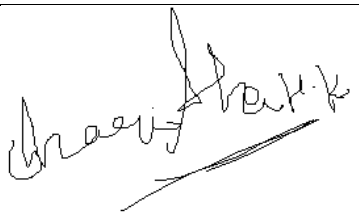
Signature of the HoD
Dr. Hemanth S R
Associate Professor & Hea
Dept. of CSE (AI)

DECLARATION BY THE STUDENTS

I/We hereby declare that the Minor Project Report titled “complaint management system”, submitted for the 5th Semester Minor Project, is a record of original work carried out by me/us under the supervision of [Guide Name].

This report has not been submitted in this or any other Institution or University. Proper acknowledgements have been made wherever the work of others has been referred to. Wherever AI tools were used for minor assistance, the same has been duly acknowledged as per academic norms.

Name and Signature of the Student(s):

	Name	Signature
1.	AFIFA TASKEEN	
2.	CHARITHA H K	

ACKNOWLEDGEMENT

We express our sincere gratitude to the Principal, Maharaja Institute of Technology Mysore, for providing a supportive academic environment and necessary facilities to carry out this minor project successfully.

We would like to extend our heartfelt thanks to the Head of the Department, Department of Computer Science and Engineering (Artificial Intelligence), for their guidance, encouragement, and continuous support throughout the course of this project.

We express our deep sense of gratitude to our internal guide, **[MS SAHANA]**, **[AST PROFESSOR]**, Department of CSE (AI), MIT Mysore, for their valuable guidance, constructive suggestions, and constant encouragement during the development of this project. Their insights and feedback were instrumental in shaping this work.

Finally, we thank all the faculty members of the Department of CSE (AI) and our peers who directly or indirectly supported us during the successful completion of this project.

AFIFA TASKEEN

CHARITHA H K

TABLE OF CONTENTS

Contents

LIST OF FIGURES

LIST OF TABLES

SMART STUDENT COMPLAINT MANAGEMENT SYSTEM

1 INTRODUCTION

1.1 Overview

With the rapid growth of digital technologies, educational institutions are increasingly adopting web-based systems to automate administrative and student-support activities. One important area that requires improvement is the handling of student complaints, which are often managed manually through registers, emails, or informal communication. These traditional methods lack transparency, are time-consuming, and do not provide proper tracking or accountability.
The Smart Student Complaint Management System aims to digitize and streamline the complaint submission and resolution process using a centralized web-based platform, ensuring efficient communication between students and administrators.

1.2 Problem Statement

In many colleges, student complaints related to academics, hostel facilities, or infrastructure are handled manually. This leads to delays, miscommunication, lack of tracking, and unresolved issues. Students are unable to monitor the status of their complaints, while administrators struggle to manage multiple requests efficiently. Hence, there is a need for a smart, transparent, and automated complaint management system.

1.3 Existing System

The existing complaint handling system relies on manual registers, emails, or verbal communication. Complaints are often misplaced, delayed, or ignored due to the absence of proper tracking mechanisms. There is no centralized database, no status monitoring, and no priority-based handling. These limitations reduce accountability and efficiency.

1.4 Proposed System

The proposed Smart Student Complaint Management System provides a centralized web-based platform where students can submit complaints online. Each complaint is automatically assigned an ID, category, priority, and timestamp. Administrators can view, update, and resolve

complaints through an admin dashboard. The system improves transparency, efficiency, and response time.

1.5 Objectives

- To provide an online platform for student complaint submission
- To ensure transparent tracking of complaint status
- To reduce manual effort and paperwork
- To prioritize complaints automatically
- To enable administrators to manage complaints efficiently

1.6 Methodology

The system is developed using a structured approach. Requirements were gathered from students and faculty. The system was designed using modular architecture. Backend APIs were developed using Node.js and Express.js, while MongoDB was used for data storage. Frontend interfaces were built using HTML, CSS, and JavaScript. The system was tested using functional and integration testing methods.

Organization of the Report

Chapter 1 introduces the project and its objectives.
Chapter 2 presents the literature survey.
Chapter 3 discusses software requirement specifications.
Chapter 4 explains system analysis and design.
Chapter 5 covers implementation details.
Chapter 6 describes testing.
Chapter 7 discusses results, followed by conclusion and future work.

2 LITERATURE SURVEY

2.1Survey Papers / Articles

- Web-Based Complaint Management Systems
  This study discusses digital platforms for complaint handling using web technologies. The system improves response time but lacks intelligent prioritization.
- Student Support Systems in Higher Education
  This article highlights the need for centralized student service platforms. However, manual intervention is still high.
- MongoDB for Scalable Web Applications
  This paper explains the use of NoSQL databases for flexible data storage. The limitation is lack of built-in analytics.
- Role-Based Access Control Systems
  Describes secure access mechanisms for multi-user platforms. Complexity increases with scale.
- Automation in Educational Administration
  Focuses on automating academic processes but does not emphasize complaint resolution.

2.2 Survey Findings

Existing systems lack transparency, automation, and real-time tracking. Most platforms do not provide priority-based complaint handling or user-friendly dashboards. The proposed system addresses these gaps by offering automation, role-based access, and centralized management.

3 SOFTWARE REQUIREMENT SPECIFICATIONS

3.1Functional Requirements

- Student registration and login
- Complaint submission with category and description
- Automatic complaint ID generation
- Admin dashboard for complaint management
- Status update functionality

Table 1 3.1Functional Requirements

Sl. No	Function	Description
1	User Registration/Login	Allows students and administrators to log in using email credentials
2	Role-Based Access	Differentiates access between student and admin users
3	Complaint Submission	Students can submit complaints with title, category, and description
4	Complaint Categorization	Complaints are categorized (Hostel, Academics, Infrastructure)
5	Priority Assignment	Complaint priority is assigned automatically
6	View Complaint Status	Students can track complaint status
7	Admin Complaint Management	Admin can view, update, and resolve complaints
8	Database Storage	Stores complaint details securely in MongoDB

The above table describes the core functionalities supported by the Smart Student Complaint Management System.

3.2 Non-Functional Requirements

Table 2 3.2 Non-Functional Requirements

Sl. No	Requirement	Description
1	Performance	System should respond within acceptable time
2	Security	User authentication and data protection
3	Usability	Simple and user-friendly interface
4	Scalability	Can handle multiple users simultaneously
5	Reliability	System should be available with minimal downtime
6	Maintainability	Easy to update and modify features

- Secure authentication
- Fast response time
- Scalable database
- User-friendly interface

Additional Requirements

- Fast response time
- Scalable database
- User-friendly interface
- Internet connectivity
- Modern web browser

3.4 System Requirements

Hardware:

- Minimum 4GB RAM
- Intel i3 or higher processor

Software**:**

- Windows OS
- Node.js
- MongoDB
- VS Code
- Web Browser

Table 3 3.4.1 System Requirements hardware

Component	Specification
Processor	Intel i3 or above
RAM	Minimum 4 GB
Hard Disk	20 GB free space
System Type	64-bit machine
Internet	Required

Software:

Table 4software 3.4.2 System Requirements

Software	Description
Operating System	Windows / Linux
Frontend	HTML, CSS, JavaScript
Backend	Node.js, Express.js
Database	MongoDB
Tools	VS Code, Cursor
Browser	Google Chrome

- Windows OS
- Node.js
- MongoDB
- VS Code
- Web Browser

4 SYSTEM ANALYSIS AND DESIGN

4.1 System Analysis

System analysis involves studying the existing complaint handling process in educational institutions and identifying its limitations. Traditional complaint systems are often manual, time-consuming, and lack transparency. Students are unsure about the status of their complaints, and administrators face difficulty in managing and tracking multiple complaints efficiently.

The proposed Smart Student Complaint Management System aims to overcome these issues by providing an automated, centralized, and role-based system. The analysis phase focuses on understanding user requirements, system functionalities, and constraints. The primary users of the system are students and administrators, each having distinct access privileges.

The system ensures proper complaint submission, categorization, tracking, and resolution, thereby improving efficiency and accountability.

4.2System Design

System design translates the requirements identified during system analysis into a structured solution. The design phase focuses on defining the system architecture, data flow, database structure, and user interactions.

The system follows a three-tier architecture consisting of:

- Presentation Layer:
  User interfaces developed using HTML, CSS, and JavaScript for students and administrators.
- Application Layer:
  Backend logic implemented using Node.js and Express.js to handle authentication, complaint processing, and status updates.
- Data Layer:
  MongoDB database used to store user details, complaint records, statuses, and timestamps.

This architecture ensures scalability, maintainability, and secure data management.

4.3 System Architecture

The system architecture illustrates the interaction between frontend, backend, and database components. Users access the system through a web interface. Requests are sent to the backend server, which processes them and communicates with the database to store or retrieve information.

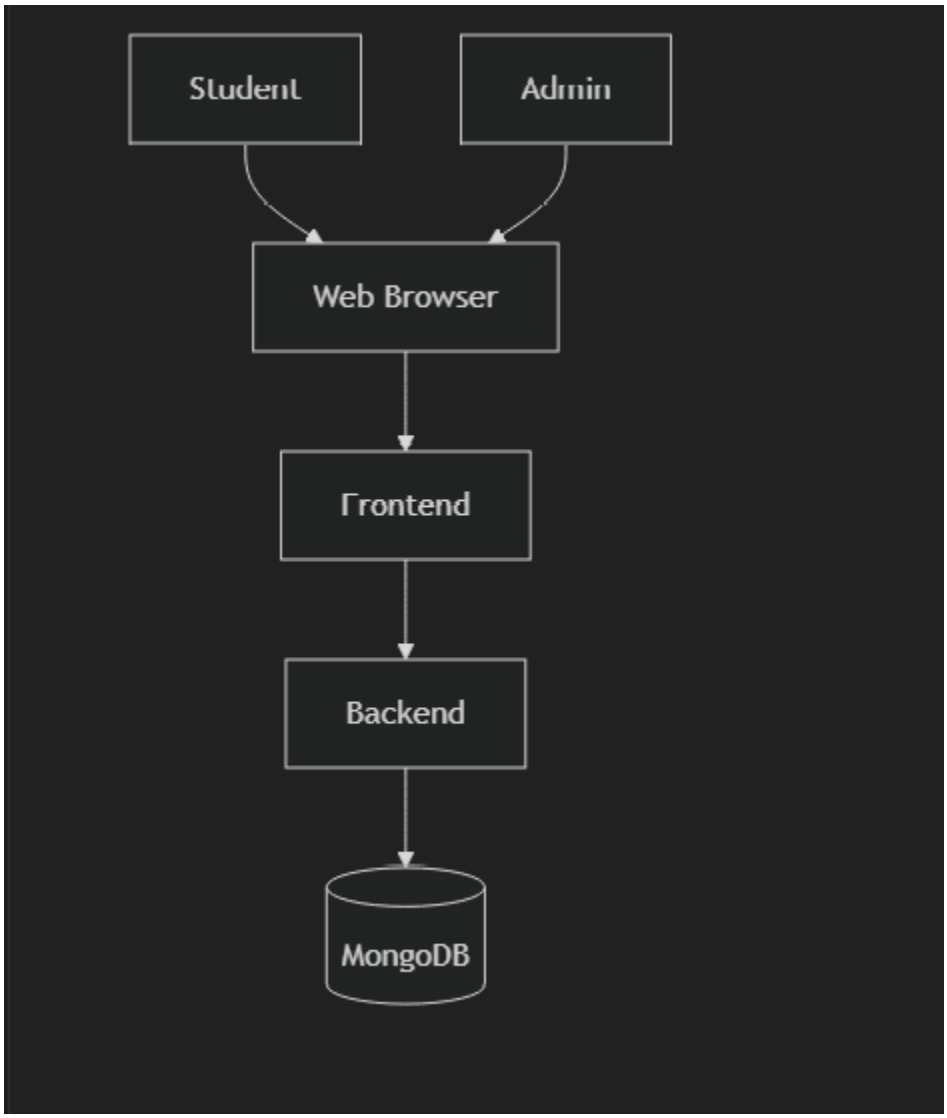


Figure 1 4.3 System Architecture Diagram of Complaint Management System

4.3 System Architecture Diagram of Complaint Management System

4.4 UML Diagrams

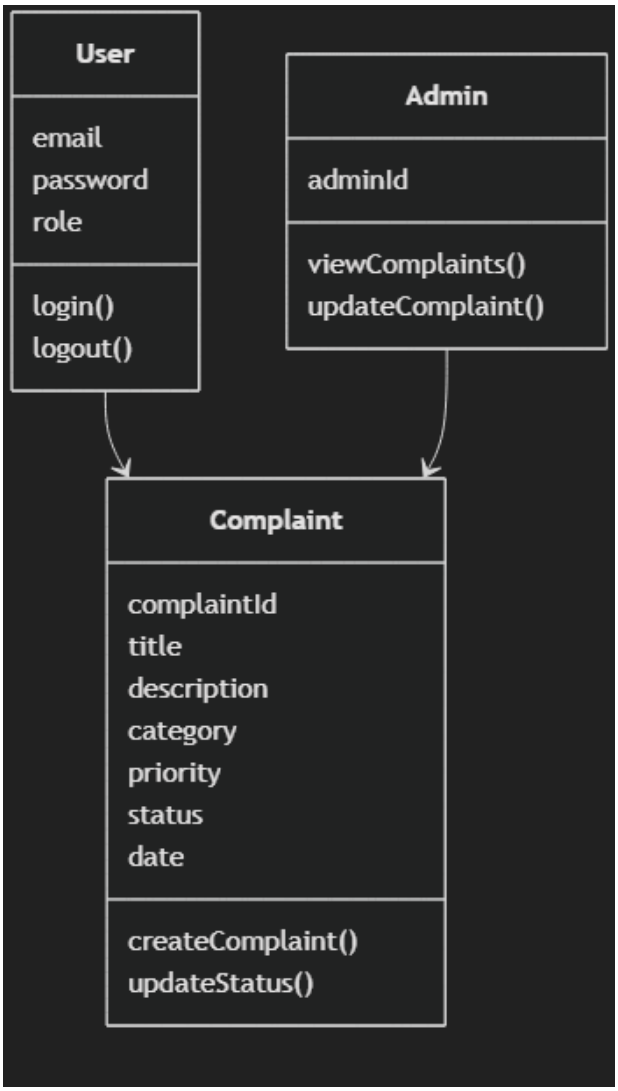


Figure 2 4.4 UML Class Diagram of Complaint Management System

4.4 UML Class Diagram of Complaint Management System

UML diagrams are used to visually represent the system design and interactions between users and system components.

4.5 Use Case Diagram

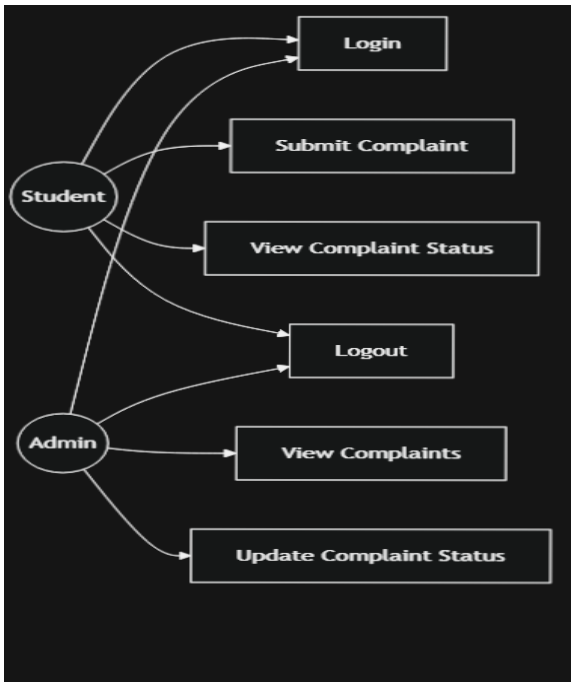


Figure 34.5 Use Case Diagram Showing Student and Admin Interactions

4.5 Use Case Diagram Showing Student and Admin Interactions

The use case diagram represents the functional requirements of the system and interactions between users and the system.

Actors**:**

- Student
- Admin

Student Use Cases:

- Login
- Submit Complaint
- View Complaint Status

Admin Use Cases**:**

- Login
- View Complaints
- Update Complaint Status

4.6 Sequence Diagram

The sequence diagram illustrates the flow of actions during complaint submission. It shows how the student submits a complaint, how the backend processes the request, and how the data is stored in the database.

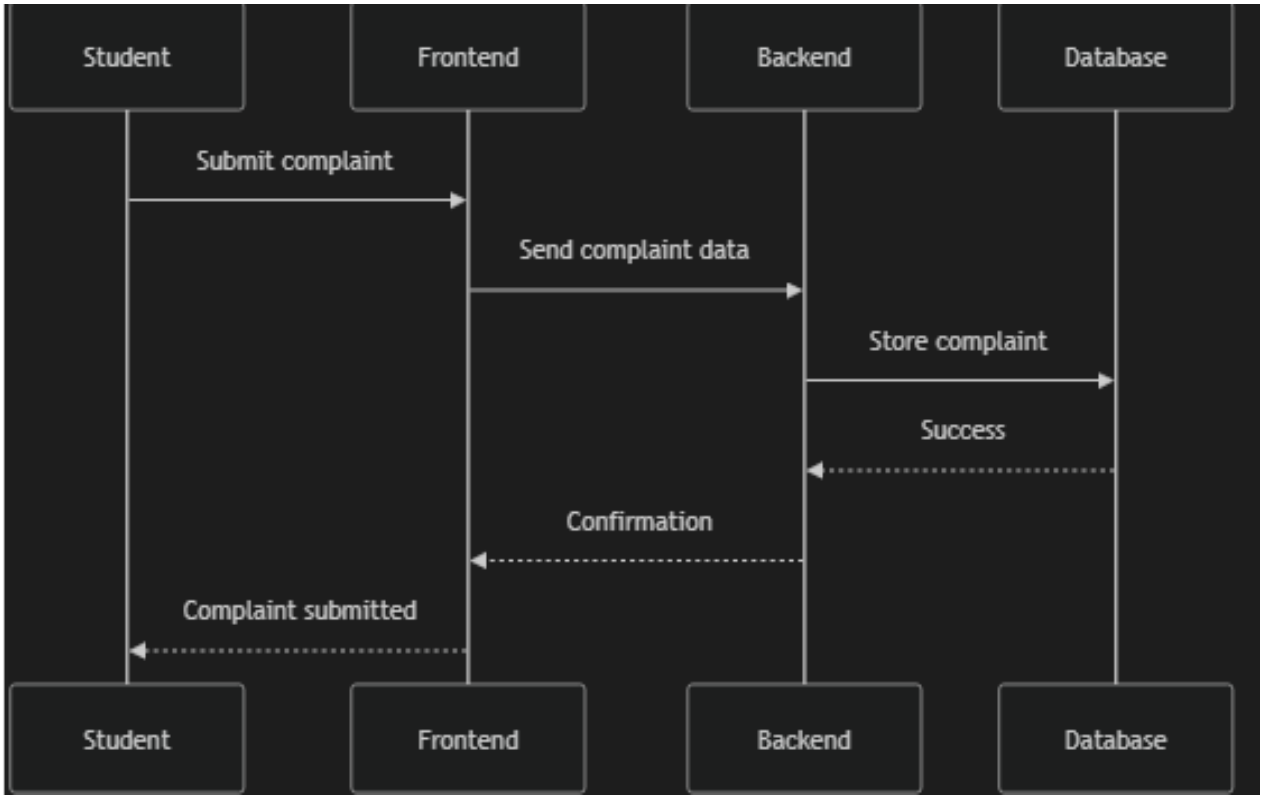


Figure 44.6 sequence Diagram for Complaint Submission Process

4.6 sequence Diagram for Complaint Submission Process

4.7 Activity Diagram

The activity diagram represents the workflow of the complaint management process from login to complaint resolution.

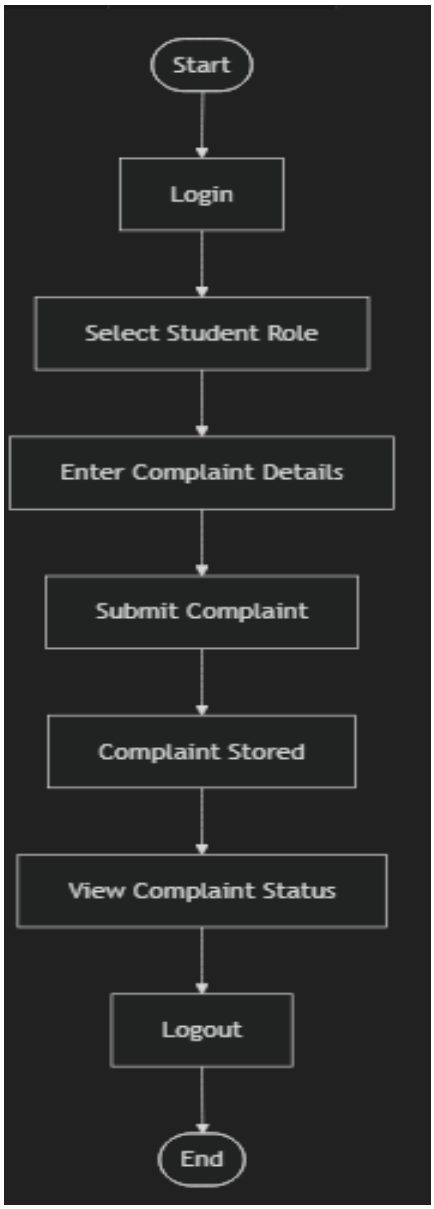


Figure 54.7 Activity Diagram for Complaint Registration and Processing

4.7 Activity Diagram for Complaint Registration and Processing

4.5 Database Design

The database design defines the structure of data storage used in the system. MongoDB is used as a NoSQL database to store complaint and user information.

The main entities include:

- User
- Complaint
- Admin

Each complaint record contains attributes such as complaint ID, title, description, category, priority, status, date, and user reference.

5 IMPLEMENTATION DETAILS

5.1 Introduction

This chapter describes the implementation details of the Smart Student Complaint Management System**.** The system was developed to provide a structured and transparent platform for students to submit complaints and for administrators to manage and resolve them efficiently. The implementation phase converts the system design into a working application using modern web technologies.

System Requirements

Table 5 3.4 System Requirements

1	**Layer**	**Technology**
2	Frontend	HTML, CSS, JavaScript
3	Backend	Node.js, Express.js
4	Database	MongoDB
5	API Testing	Postman
6	Development Tool	VS Code
7	Browser	Google Chrome

5.3 Frontend Implementation

The frontend provides the user interface for students and administrators.

Student Module

- Login interface
- Complaint submission form
- View complaint status
- View complaint details (date, category, priority, status)

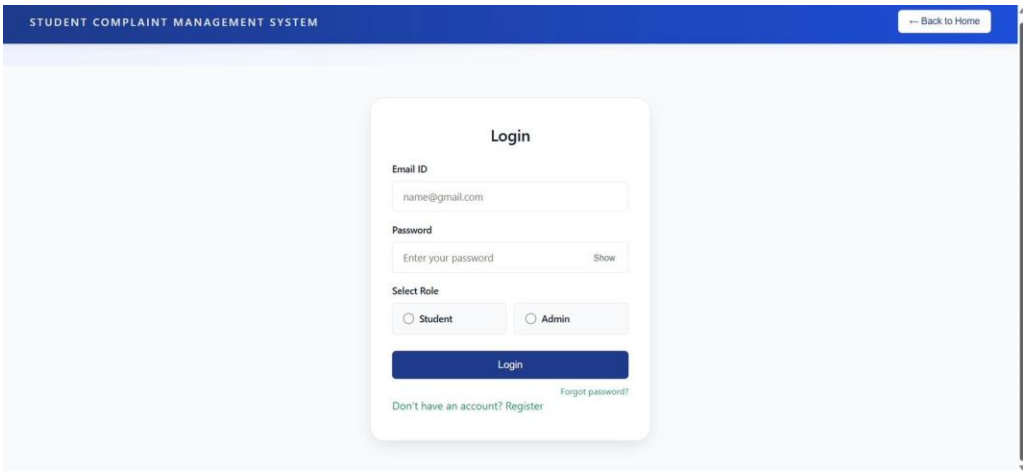


Figure 65.3 Login Page Interface of Complaint Management System

5.3 Login Page Interface of Complaint Management System

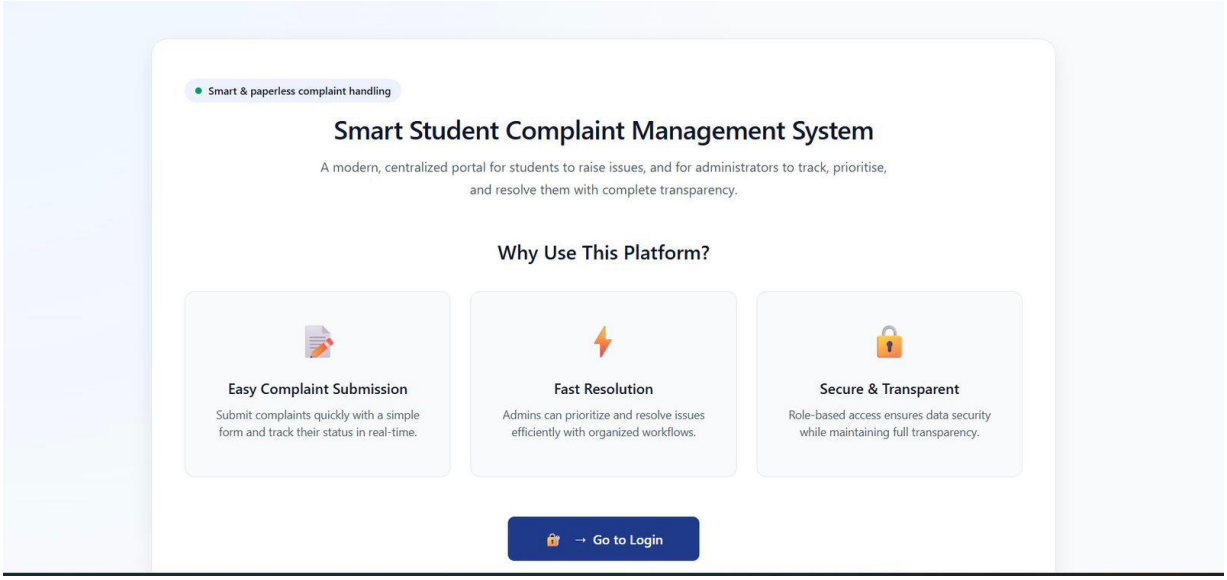


Figure 75.3 Dashboard of complaint management system

5.3.1 Dashboard of complaint management system

Admin Module

- Secure admin login
- View all complaints
- Update complaint status
- View complaint description and details

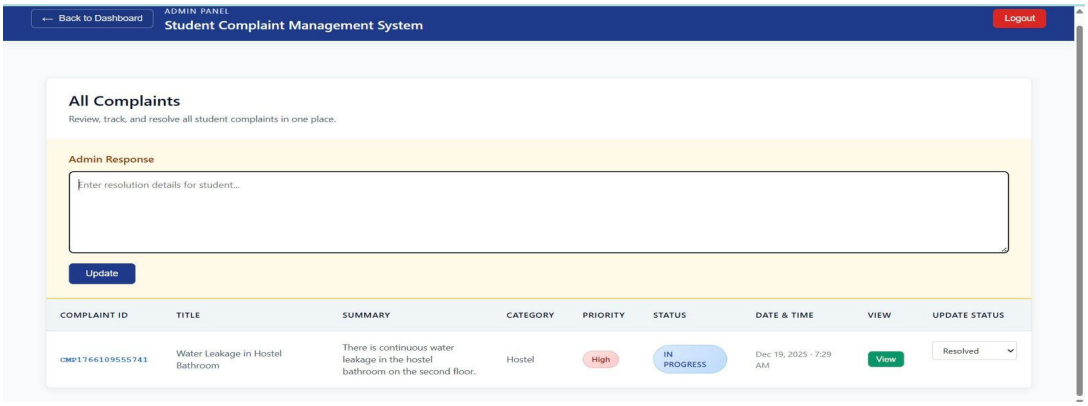


Figure 8 5.3.2 Admin Dashboard for Complaint Monitoring and Management

5.3.2 Admin Dashboard for Complaint Monitoring and Management

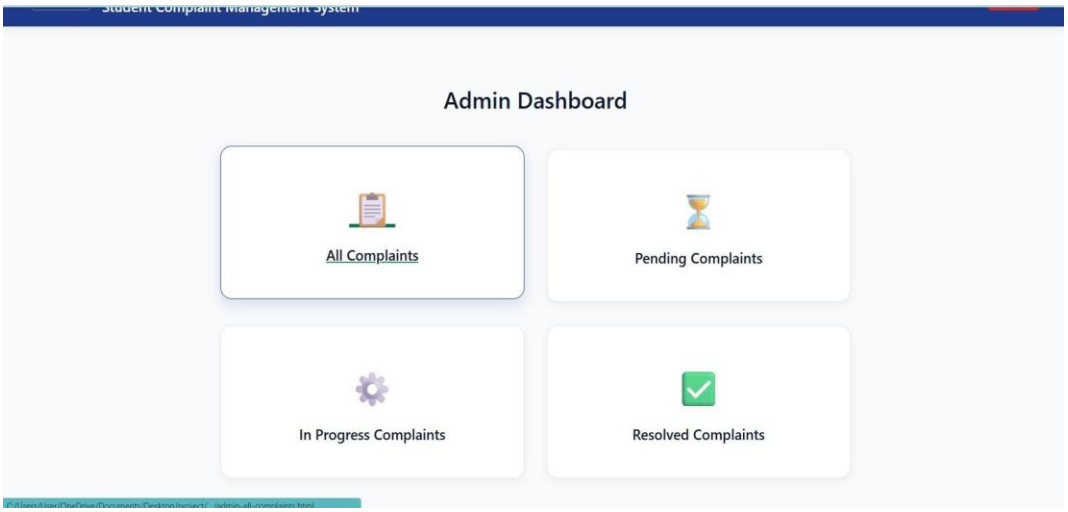


Figure 9 5.3.3 admin dashboard

5.3.3 Admin Dashboard for Complaint Monitoring and Management

Backend Implementation

The backend was developed using Node.js and Express.js. It handles:

- Request processing
- Business logic
- Communication with MongoDB

Key Backend Features

- Restful APIs for complaint submission
- Complaint retrieval for students and admin
- Status update functionality
- Unique complaint ID generation
- Role-based access handling

Key API Endpoints

- POST /complaints – Submit a new complaint
- GET /complaints – Retrieve all complaints
- GET /complaints/student – Retrieve complaints by student
- PUT /complaints/:id – Update complaint status

5.4 Database Implementation

MongoDB is used to store complaint-related data in a structured document format.

Complaint Collection Structure

Table 6 5.4 Database Implementation

Field Name	**Description**
complaint ID	Auto-generated unique ID
title	Complaint title
description	Detailed complaint
category	Hostel / Academics / Infrastructure
priority	Low / Medium / High
status	Pending / In Progress / Resolved
date	Date of submission
createdByEmail	Student email

Complaint Workflow Implementation

- Student logs in
- Student submits complaint
- Complaint stored in MongoDB
- Admin views complaint
- Admin updates status
- Student views updated status

Security Measures

- Input validation
- Role-based page access
- Restricted admin actions
- Secure API handling

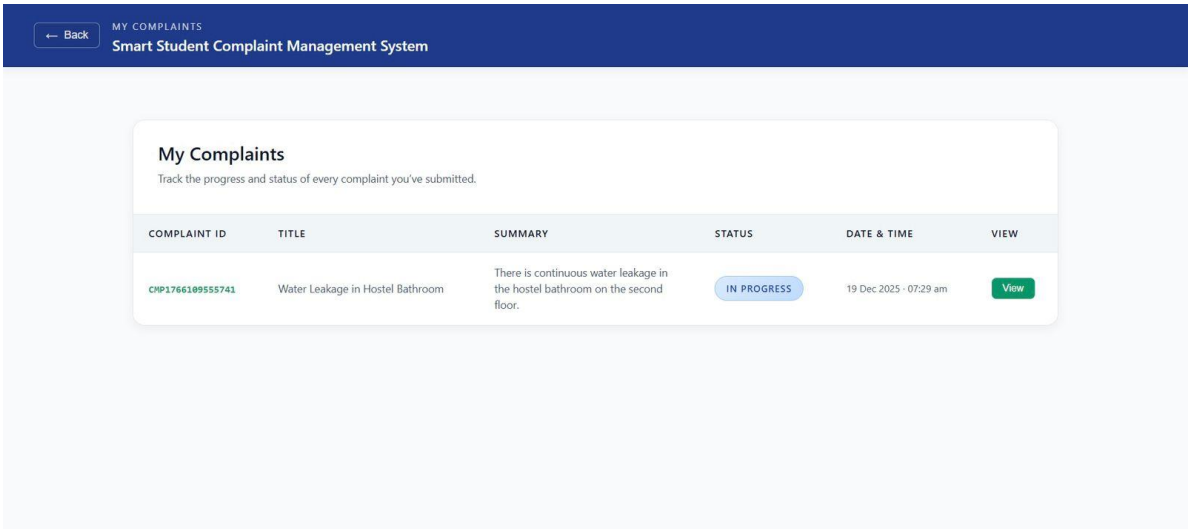


Figure 105.4 Student Complaint Status and History Page

5.4 Student Complaint Status and History Page

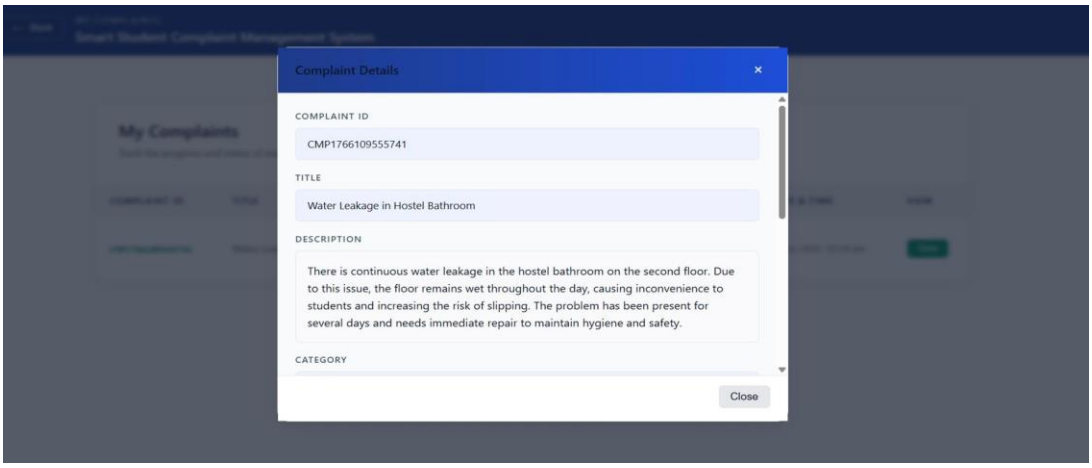


Figure 115.5 Detailed Complaint View in Admin Panel

5.5 Detailed Complaint View in Admin Panel

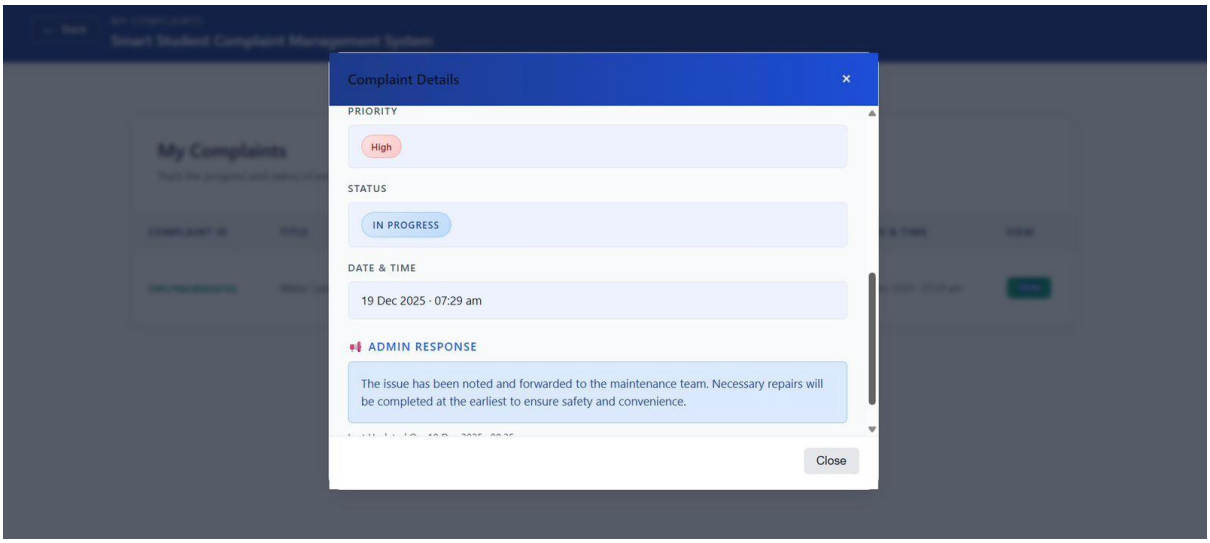


Figure 125.5.1 Detailed Complaint View in Admin Panel

5.5 Summary

The implementation successfully integrates frontend, backend, and database components to create a reliable and efficient complaint management system. The system ensures transparency, accountability, and improved communication between students and administrators.

6 TESTING DETAILS

6.1 Introduction to Testing

Testing is an essential phase of software development that ensures the system functions correctly according to the specified requirements. The objective of testing is to identify defects, verify data accuracy, validate user workflows, and ensure the reliability of the application.
In the Smart Student Complaint Management System, testing was conducted to verify complaint submission, storage in MongoDB, role-based access, complaint viewing, status updates, and overall system performance.

6.2 Testing Methodology

The following testing methodologies were adopted:

- Manual Testing
- Functional Testing
- Integration Testing
- System Testing
- User Acceptance Testing

Each module was tested independently and later tested as a complete system.

6.3 Types of Testing Performed

Unit Testing
Unit testing was performed on individual modules to verify that each function operates correctly.
Modules Tested

- Login Module
- Complaint Submission Module
- Complaint Viewing Module
- Admin Dashboard Module
- Status Update Module

Unit Testing Test Cases

Table 7 6.3.1 Unit Testing Test Cases

Test Case ID	Module	Test Description	Input	Expected Output	Status
UT01	Login	Validate user login	Valid credentials	Login successful	Pass
UT02	Complaint Form	Submit complaint	Valid data	Complaint stored	Pass
UT03	View Complaints	Fetch complaints	Student login	Complaints displayed	Pass
UT04	Admin Dashboard	Load dashboard	Admin login	Dashboard visible	Pass
UT05	Status Update	Change status	New status	Status updated	Pass

Integration Testing

Integration testing ensures that different modules interact correctly with each other and the database.

Integration Testing Test Cases

Table 8 6.3.2 Integration Testing Test Cases

Test Case ID	Integrated Modules	Description	Expected Result	Status
IT01	Login + Dashboard	Role-based redirection	Correct dashboard loads	Pass
IT02	Complaint + Da tabase	Save complaint	Stored in MongoDB	Pass
IT03	Admin + Database	Fetch complaints	All complaints displayed	Pass
IT04	Status Update + DB	Update complaint	Status reflected	Pass

System Testing

System testing was carried out to validate the complete end-to-end functionality of the system.

System Testing Test Cases

Table 9 6.3.3 System Testing Test Cases

Test Case ID	Scenario	Expected Output	Status
ST01	Student submits complaint	Complaint saved	Pass
ST02	Student views status	Status displayed	Pass
ST03	Admin updates complaint	Status updated	Pass
ST04	Logout and re-login	Session handled	Pass

User Acceptance Testing (UAT)

User Acceptance Testing was performed by students and admin users to ensure the system meets real-world usage requirements.

User Acceptance Test Results

Feature	User Feedback	Acceptance Status
Complaint Submission	Easy to use	Accepted
Complaint Tracking	Useful	Accepted
Admin Dashboard	Clear layout	Accepted
Status Updates	Effective	Accepted

6.4 Test Environment

Test Environment Details

Table 10 6.4 Test Environment Details

Component	Specification
Operating System	Windows 10
Browser	Google Chrome
Backend	Node.js + Express
Database	MongoDB
Hardware	Intel i5, 8GB RAM

6.5 Error Handling and Debugging

During testing, issues such as incorrect routing, missing status updates, and form validation errors were identified. These issues were resolved by correcting API routes, improving database queries, and adding client-side validations.

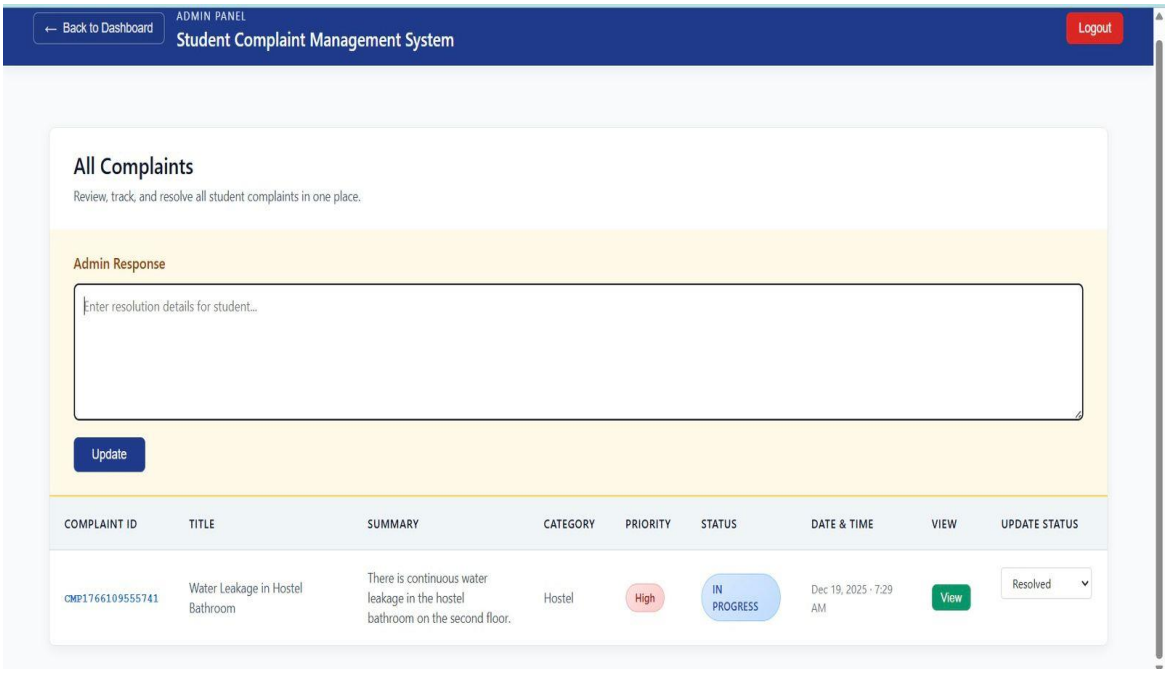


Figure 136.5 Admin dashboard complaint list

6.5 Admin dashboard complaint list

6.6 Testing Summary

All modules of the Smart Student Complaint Management System were tested successfully. The system reliably handles complaint submission, storage, retrieval, and status updates. Testing confirms that the application meets its functional requirements and is ready for deployment.

7 RESULTS AND DISCUSSION

7.1 Snapshots

Screenshots of login page, student dashboard, and admin dashboard are included.

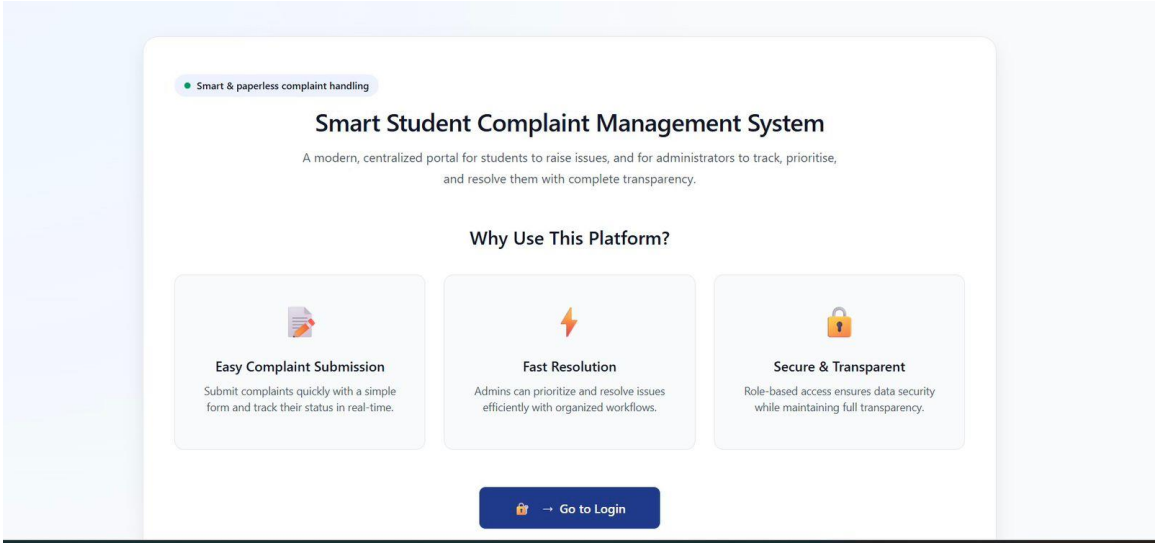


Figure 14 7.1 snapshot of login page

7.1 snapshot of login page

Result Discussion

7.2 Introduction

This chapter presents the results obtained after implementing and testing the Smart Student Complaint Management System**.** It discusses how the system performs in real-time scenarios, evaluates the effectiveness of the implemented features, and highlights the improvements achieved over the traditional manual complaint handling process.

System Output

After successful implementation, the system produces the following outcomes:

Student Module Results

- Students are able to register and log in successfully.
- Complaint submission is smooth and validated.
- Complaints are stored securely in the MongoDB database

Students can view:

- Complaint title
- Category
- Priority
- Status
- Date and time of submission

Admin Module Results

- Admin can log in securely.
- All complaints submitted by students are displayed in a structured table.
- Admin can:
- View full complaint descriptions
- Update complaint status (Pending, In Progress, Resolved)
- Changes made by admin are reflected immediately for students.

7.3 Database Results

- All complaints are stored as documents in MongoDB.
- Each complaint record contains:
- Complaint ID
- Student email
- Title
- Description
- Category
- Priority
- Status
- Date and time
- Data retrieval using APIs is fast and consistent.

7.4 Performance Analysis

Table 11 7.4 Performance Analysis

Parameter	Observation
Response Time	Fast for both submission and retrieval
Data Accuracy	No data loss observed
System Reliability	Stable during multiple submissions
User Interface	Simple, responsive, and user-friendly

7.5 Advantages of the System

- Eliminates manual paperwork
- Faster complaint resolution process
- Transparent complaint tracking
- Secure data storage
- Easy to maintain and extend

7.6 Limitations

- No role-based authentication (email verification) implemented yet
- No notification system for status updates
- AI-based analysis is not included in the current version

7.7 Discussion

The results show that the Smart Student Complaint Management System successfully meets its objectives. The system improves communication between students and administration, reduces delay in complaint handling, and provides a structured and transparent workflow.

8 CONCLUSION AND FUTURE WORK

8.1 Conclusion

The Smart Student Complaint Management System was successfully designed and implemented to provide an efficient, transparent, and user-friendly platform for handling student complaints within an educational institution.

The system replaces the traditional manual complaint process with a digital solution that allows students to submit complaints online and track their status in real time. The admin module enables authorities to view, manage, and resolve complaints efficiently. The use of modern web technologies and MongoDB ensures reliable data storage, faster processing, and scalability.

Overall, the project achieves its primary objective of improving complaint handling efficiency, accountability, and communication between students and administration.

Achievements of the Project

- Successful implementation of a web-based complaint management system
- storage of complaint data using MongoDB
- Secure Real-time complaint status updates
- Separate dashboards for students and administrators
- Improved transparency and reduced response time
- User-friendly and responsive interface

8.2 Future Enhancements

Although the current system is functional and efficient, several enhancements can be implemented in the future to improve performance and usability:

Authentication and Authorization

- Implement email-based authentication
- Role-based access control for students and admins
- Password encryption and secure login

Notification System

- Email or SMS notifications for complaint status updates
- Alerts for admins when new complaints are submitted

AI-Based Features

- Automatic complaint categorization using AI
- Priority prediction based on complaint content
- Duplicate complaint detection

Analytics and Reports

- Complaint statistics dashboard
- Graphical reports for admin decision-making
- Monthly and category-wise analysis

Mobile Application

- Android/iOS app for easier access
- Push notifications for updates

8.3 Final Remarks

The Smart Student Complaint Management System is a practical, scalable, and industry-relevant project suitable for academic evaluation. With further enhancements, it can be deployed in real-world institutions to significantly improve grievance handling mechanisms.