

# **Online Complaint Management System**

**23MCCE11,23MCCE15**

## **Feasibility Study**

A feasibility study is carried out to analyse whether the proposed Online Complaint Management System can be developed and implemented successfully. This study helps in understanding the practicality of the system in terms of technology, cost, operation, and time constraints before actual development begins.

---

### **1 Technical Feasibility**

Technical feasibility examines whether the required technology, tools, and technical skills are available to develop the proposed system.

The Online Complaint Management System is developed using well-established and widely supported technologies. The backend is implemented using **Spring Boot**, which provides a robust framework for building secure and scalable web applications. **Spring MVC** is used for handling requests and responses, while **Spring Security** ensures secure authentication and session management. The frontend uses **HTML, CSS, and JavaScript**, which are standard web technologies and easy to implement. **MySQL** is used as the database to store user details, complaints, and complaint status information. The application runs on an **embedded Tomcat server**, eliminating the need for external server configuration.

All the required software tools are open-source and easily available. The development team also has sufficient knowledge of Java and web technologies. Hence, the system is technically feasible.

### **2 Economic Feasibility**

Economic feasibility evaluates whether the system is cost-effective and affordable.

The Online Complaint Management System does not require any expensive hardware or licensed software. All the technologies used, such as Spring Boot, MySQL, and frontend tools, are free and open-source. Since this is an academic

mini-project, the development cost is minimal. The system reduces manual effort and paperwork, which further helps in lowering operational costs.

Therefore, the project is economically feasible and suitable for implementation within limited budget constraints.

### **3 Operational Feasibility**

Operational feasibility checks whether the proposed system will function effectively in a real-world environment and whether users will be comfortable using it.

The system provides a simple and user-friendly interface for complaint submission and tracking. Users can easily register, log in, submit complaints, and track their status without any technical difficulty. Administrators can efficiently review complaints and update their status through a secure admin panel. The centralized complaint management approach improves transparency and reduces delays in complaint handling.

Minimal training is required for both users and administrators, making the system easy to operate. Hence, the system is operationally feasible.

### **4 Schedule Feasibility**

Schedule feasibility determines whether the project can be completed within the given time frame.

The project is divided into well-defined modules such as user authentication, complaint submission, complaint tracking, admin management, and reporting. The scope of the project is limited and clearly defined, making it manageable within the academic semester. The use of existing frameworks like Spring Boot speeds up development and reduces implementation time.

Therefore, the Online Complaint Management System can be completed within the allotted schedule.

## **Functional Requirements**

### **User Registration and Login**

The system should allow new users to create an account using basic details such as name, email, phone number, and password. The email must be unique so that each account belongs to only one user. Once registered, users should be able to log in securely using their credentials. After successful login, a session must be created so that the user can access complaint-related features without logging in repeatedly. The system should also allow users to log out, which will terminate the session.

### **Complaint Submission**

A logged-in user must be able to submit a complaint through an online form. The complaint should include a title, category, and description explaining the issue clearly. The system should validate all required fields before accepting the submission. Once submitted, the complaint should be stored in the database and a unique reference ID should be generated automatically. By default, every new complaint should be marked as “Pending.”

### **Complaint Tracking**

Users should be able to track the status of their submitted complaints at any time. After logging in, they must be able to see a list of their complaints along with details such as submission date, current status, and any remarks added by the administrator. Users must not be able to view complaints submitted by others.

### **Admin Management**

The system should provide a separate login for administrators. After authentication, the admin should be able to view all complaints in the system. The admin must have the authority to update complaint status (such as Pending, In Progress, or Resolved) and add remarks or resolution notes. Any updates made by the admin should immediately reflect in the user's tracking view.

### **Basic Reporting**

The system should allow the administrator to view a simple summary of complaints, such as the total number of complaints, how many are pending,

and how many are resolved. This feature is meant for monitoring purposes and is limited in scope for the mini-project version.

## **Non-Functional Requirements**

### **Performance**

The system should respond to user actions within a few seconds under normal usage. Since this is a mini-project, it is expected to handle a moderate number of users without noticeable delay.

### **Security**

User passwords must be stored in encrypted form in the database to protect sensitive information. Only authenticated users should be able to submit and track complaints, and only authorized admins should be able to update complaint status. The system must prevent unauthorized access and basic security threats such as SQL injection.

### **Usability**

The interface should be simple and easy to understand, even for users who are not technically skilled. Forms should be clearly labelled, and error messages should be meaningful and helpful instead of technical. The system should work properly on commonly used web browsers.

### **Reliability and Data Integrity**

Once a complaint is submitted, it should not be lost or altered unintentionally. The database should maintain consistency and ensure that all complaint records remain accurate. The system should function properly as long as the server is running and internet connectivity is available.

### **Maintainability and Scalability**

The application should follow a clean Spring Boot architecture so that it can be easily maintained or upgraded in the future. New features such as email notifications, graphical reports, or additional user roles should be possible to add without redesigning the entire system.