# CHAPTER 4

## ARCHITECTURAL DESIGN

The architecture of the Paperless Student Complaint System is designed to streamline the process of complaint registration, tracking, and resolution in a fully digital environment. The system follows a structured pipeline that ensures data validation, secure submission, and efficient complaint management by students and administrators.

1. User Interface (Frontend)  
- Students and Admins access the web portal through an intuitive user interface built with HTML, CSS, and Flask templates.  
- The interface allows login, complaint submission, viewing complaint status, and admin response.

2. Authentication & Authorization  
- Student Login: Verified via registration number, department, year, username, and password.  
- Admin Login: Admins authenticate with a secure ID and password.  
- Role-based access controls prevent unauthorized actions (e.g., only admins can update status).

3. Complaint Submission Module  
- After successful login, students can fill out a complaint form with fields like:  
 - Complaint Title  
 - Description  
 - Category (e.g., hostel, staff, exam, canteen)  
 - Date of Submission  
- Data is validated client-side and server-side before processing.

4. Complaint Processing & Storage  
- On submission, complaints are stored in a MySQL database using flask\_mysqldb.  
- Each complaint is tagged with a unique ID and linked to the student’s account via their email or registration number.

5. Admin Dashboard Module  
- Admins can view all submitted complaints categorized by status:  
 - New  
 - In Progress  
 - Resolved  
- Admins can update the status and provide feedback, which is instantly reflected in the student’s dashboard.

6. Complaint Status Tracker  
- Students can track their complaint status in the “My Complaints” page.  
- They receive real-time updates and response history related to each complaint.

7. Notifications & Feedback  
- Flash messages notify users of successful logins, submission, or updates.  
- Optional: Email alerts or SMS can be added in future enhancements.

8. Database Design  
- Key tables:  
 - users: Stores student login credentials and verification info.  
 - complaints: Contains all complaint records, including timestamps, status, and admin responses.  
 - admins: Stores admin credentials.

9. Security Measures  
- Passwords are hashed using werkzeug.security.  
- Sessions are managed securely using Flask session cookies and secret keys.  
- Inputs are sanitized to prevent SQL injection or XSS attacks.

10. Deployment Environment  
- The application is hosted on a local server using WAMP or Flask’s development server.  
- It can be deployed online using platforms like Render, Heroku, or PythonAnywhere for wider access.