Software Testing Plan

College Complaint Management System

1. Introduction

This Software Testing Plan is prepared for the College Complaint Management System based on the provided case study. The document follows the same structure and headings as the sample testing document. The objective of this plan is to define the testing strategy, test environment, test cases, and defect management process to ensure system quality.

2. Test Strategy

The testing strategy for the College Complaint Management System includes multiple levels of testing:

• Unit Testing – Individual modules such as user authentication, complaint submission, and status updates are tested separately to verify correctness.

• Integration Testing – Interaction between modules such as complaint workflow, notification system, and database operations is tested.

• System Testing – The complete system is tested to validate functional and non-functional requirements.

• Acceptance Testing – Conducted with end users and administrators to ensure the system meets institutional requirements.

3. Test Environment

The test environment is designed to simulate the actual operating environment of the system.

• Hardware Requirements – Desktop or laptop systems with internet connectivity.
• Software Requirements – Web browsers, backend server environment, and database server.
• Test Data – Sample student records, complaint details, department data, and user credentials.

This environment ensures accurate and reliable testing outcomes.

4. Test Cases

Test cases are designed to validate system functionality.

Example Test Cases:

• Login Test – Valid and invalid user credentials; expected result is successful login or error message.
• Complaint Submission Test – Submission of complaint details; expected result is complaint ID generation.
• Status Tracking Test – Viewing complaint status; expected result is real-time status display.
• Notification Test – Status update notification; expected result is alert delivery.

Each test case includes defined input data, expected output, and pass/fail criteria.

5. Defect Management

Defect management defines the process of identifying, logging, tracking, and resolving defects.

• Defects are logged with unique identifiers and detailed descriptions.
• Severity and priority are assigned based on impact.
• Developers resolve defects and testers verify fixes.
• Closed defects are documented for future reference.

This process ensures continuous improvement and system reliability.