Software Testing On Hostel Management System

1) Sanity Testing

Sanity testing ensures that the core functionalities of the system work correctly after minor changes or bug fixes. It includes:

- Verifying **login functionality** for students and admins, ensuring successful authentication and redirection to their dashboards.
- Testing **student registration**, confirming new users can create an account and access their profiles.
- Validating the **student dashboard**, ensuring it displays attendance, fee status, room information, and complaints section accurately.
- Checking the **complaint** submission feature, ensuring complaints are stored properly and accessible to the admin.
- Verifying **admin dashboard functionalities**, such as updating room availability and managing student records, ensuring data consistency.

2) Security Testing

Security testing ensures that the Hostel Management System is **protected from vulnerabilities** and unauthorized access by:

- Preventing SQL Injection, ensuring that invalid inputs do not expose database information.
- Verifying **password encryption**, confirming that credentials are securely stored.
- **Checking session timeout**, ensuring automatic logout after prolonged inactivity.
- Restricting unauthorized access, preventing non-authenticated users from accessing admin or student dashboards.
- Testing input validation, ensuring fields do not accept script injections to protect against malicious attacks.

3) Stress Testing

Stress testing evaluates system **stability under extreme conditions** by:

- **Testing multiple concurrent logins**, ensuring the system can handle a high number of simultaneous student/admin logins without crashing.
- Submitting **bulk complaints** simultaneously, checking system response and load-handling capability.
- Validating large file uploads, ensuring proper error handling and preventing crashes.
- **Simulating low-memory conditions**, ensuring the system remains functional under resource constraints.
- Checking for **infinite loop** scenarios, ensuring system stability and error recovery mechanisms.

4) Performance Testing

Performance testing measures the system's **speed, responsiveness, and stability** under expected workloads by:

- **Testing login response time**, ensuring users can access their accounts quickly.
- **Validating page load speed**, ensuring smooth navigation for student and admin dashboards.
- **Checking complaint submission and fee report generation**, ensuring fast processing times.
- Measuring database query execution time, ensuring efficient search and retrieval of student records.