Performance Testing Report

Performance testing is a critical phase in the software testing process that ensures the system's responsiveness, stability, scalability, and speed under various workloads. The main objective is to identify and eliminate performance bottlenecks to provide a smooth and efficient user experience.

Objectives of Performance Testing:

- To determine whether the system meets performance criteria.
- To identify bottlenecks and limitations under load.
- To ensure stability under peak usage conditions.
- To evaluate scalability for future growth.
- To verify reliability and response time across environments.

Types of Performance Testing:

1. Load Testing – Determines system behavior under expected load.
2. Stress Testing – Evaluates system performance under extreme conditions.
3. Spike Testing – Tests how the system handles sudden traffic spikes.
4. Endurance Testing – Verifies system stability during long-duration operations.
5. Scalability Testing – Checks system capability when scaled up or down.

Common Tools Used:

- Apache JMeter
- LoadRunner
- Gatling
- NeoLoad
- BlazeMeter

Performance Testing Process:

1. Identify the test environment.
2. Define performance acceptance criteria.
3. Plan and design test scenarios.
4. Configure test environment and tools.
5. Execute the performance tests.
6. Analyze results and identify bottlenecks.
7. Optimize and retest to validate improvements.

Results and Observations:

During the testing process, the application demonstrated stable response times under expected loads. Minor latency was observed when the number of concurrent users exceeded the threshold, which was later optimized by improving backend query efficiency and server-side caching.

Conclusion:

Performance testing confirmed that the system is efficient, scalable, and reliable for production use. Regular monitoring and optimization are recommended to maintain performance as user demand increases.