Check *0/60 completed*

✓	Title	Task
☐	Performance Test Objectives.	What we are going to achieve.
☐	Project Scope.	What is the scope of project, example: Number of scripts, how long we need to test, Etc.
☐	Application Architecture.	Application details such app server, DB server, you can include architectural diagram if you have it.
☐	Environment Details.	Details about the environment we are going to test. It's always good to have an isolated environment for performance testing.
☐	Infrastructure Setup.	Initial setup for the performance testing (for example, cloud environment, tool installation, etc.).
☐	Performance Test Approach.	How we're going to carry out the test. We should start with a baseline test with a smaller number of users, and then gradually we can increase the users and perform different type of tests like stress, endurance, etc.
☐	Entry and Exit Criteria.	This is very important. We should always start performance testing when there are zero functional defects. Same way we should document when we can stop performance testing.
☐	Defect Management.	We should follow same tool and practices followed by client to log defect related to performance testing.
☐	Roles and Responsibilities.	Details about stake holders involved in the different activities during performance testing.
☐	Assumptions and Risks.	If there are objectives which can be a risk to performance testing, we should document it.
☐	Test Data Strategy.	Details about test data strategy and how can we extract it
☐	Test Plan Timeline and Key Deliverables.	Timeline of scripting, test execution, analysis, and deliverables for client review.
☐		
☐		
	Title	**Workload Modeling**
☐	Workload Model 1.	It's just a simple model, where number of users will be increased continuously as the test progress. Example: one user per second until the test is completed.
☐	Workload Model 2.	In this model, the number of users will be increased like a step for entire duration of the test. For example, the first 15 minutes will be 100 users and next 15 minutes will be 200, etc. We can plan this type of test for endurance testing.
☐	Workload Model 3.	This is the most common performance testing model. Number of users will be continuously increased for certain time (We call this the ramp up period). After that, users will have steady state for certain duration. Then users will start ramp down and test will finish. For example, if we're planning 1.5 hours of testing, we can give 15 minutes for ramping up the users and 15 minutes for ramp down. Steady state will be one hour. When we analyze the results, we will take only steady state for consideration.
☐	Workload Model 4.	In this model, the number of users will be increased and decreased suddenly for entire duration. There are different names for this type of testing, like monkey testing, spike testing, etc.
☐		
☐		
	Title	**Load Testing CheckList**
☐	Performance on normal usage	The performance of an application is checked with respect to its response to the user request and its ability to respond consistently within an accepted tolerance on different user loads.
☐	Performance on peak usage	The performance of an application is checked with respect to its response to the user request and its ability to respond consistently within an accepted tolerance on different user loads.
☐	Maximum load the application	What is the maximum load the application is able to hold before the application starts behaving unexpectedly?
☐	Data the Database able to handle	How much data the Database able to handle before the system slows or the crash is observed?
☐	Network related issues	Are there any network related issues to be addressed?
☐		
	Title	**Stress Testing CheckList**
☐	Stability & reliability	Stability & reliability of software application
☐	festival time	During festival time, an online shopping site may witness a spike in traffic, or when it announces a sale.
☐	Abnormal conditions.	To check whether the system works under abnormal conditions.
☐	Error message	Displaying appropriate error message when the system is under stress.

Check			0/60 completed
✓	Title	Task	
☐	System failure	System failure under extreme conditions could result in enormous revenue loss	
☐	recoverability	The main purpose of stress testing is to make sure that the system recovers after failure which is called as recoverability.	
☐	Pages per Second	Measures how many pages have been requested / Second	
☐	Throughput	Basic Metric – Response data size/Second	
☐	Rounds	Number of times test scenarios have been planned Versus Number of times a client has executed	
☐	Hit time	Average time to retrieve an image or a page	
☐	Time to the first byte	Time is taken to return the first byte of data or information	
☐	Page Time	Time is taken to retrieve all the information in a page	
☐			
	How to do Stress Testing?		
☐	Planning the Stress Test.	Here you gather the system data, analyze the system, define the stress test goals	
☐	Create Automation Scripts:	In this phase, you create the Stress testing automation scripts, generate the test data for the stress scenarios.	
☐	Script Execution:	In this stage, you run the Stress testing automation scripts and store the stress results.	
☐	Results Analysis	In this stage, you analyze the Stress Test results and identify bottlenecks.	
☐	Tweaking and Optimization	In this stage, you fine-tune the system, change configurations, optimize the code with goal meet the desired benchmark.	
☐			
	Title	Volume Testing CheckList	
☐		Test to check if there is any data loss	
☐		Check the system's response time	
☐		Check if the data is stored correctly or not	
☐		Verify if the data is overwritten without any notification	
☐		Check for warning and error messages, whether it comes at all for volume problems	
☐		Check whether high volume data affects the speed of processing	
☐		Does system have the necessary memory resources	
☐		Does volume test executed on the whole system	
☐		Is there any risk if data volume is greater than specified	
☐		Is there any guarantee that no larger date volume will occur than specified	
	Best practices for high volume testing		
☐		Stop all servers and check all logs	
☐		Before the load test manually execute the application scenario	
☐		For most useful results stagger the number of users	
☐		To overcome license constraints, balance think time	
☐		Be cautious with the new build	

Check			0/60 completed
✓	Title	Task	
☐		Analyze the use case for improvement once a baseline has been established	
☐		A repetition of particular parts of volume testing becomes inevitable in case there is a performance bottleneck	
	Resources	1.https://www.loadview-testing.com/blog/load-testing-preparation-checklist/ 2.https://www.guru99.com/load-testing-tutorial.html 3.https://www.guru99.com/stress-testing-tutorial.html	