### **Case Study: Performance Testing for Web Application - https://phptravels.org/login**

#### Objectives:

* Load Testing: Simulate a realistic number of users to evaluate the system's performance under varying loads.
* Stress Testing: Determine the system's breaking point by gradually increasing the load until performance degrades.
* Response Time Analysis: Measure and analyze response times for critical operations to identify potential bottlenecks.
* Scalability Assessment: Evaluate how the system scales with increasing user loads.

#### Steps:

1. Identify Scenarios:

* Understand the Application.
* Identify the scenarios to be performance Tested.
* Record the scenario using Jmeter.

2. Create JMeter Test Plan:

* Add Thread Groups to simulate different user profiles (e.g., developers, project managers).
* Configure HTTP Request Samplers for each scenario.
* Include HTTP Header Managers for necessary headers.
* Add listeners (e.g., View Results Tree, Summary Report) for result analysis.

3. Load Testing:

* Simulate a gradual increase in the number of users accessing the system.
* Monitor response times and server resources (CPU, memory, etc.).
* Identify the point at which the system performance starts to degrade.

4. Stress Testing:

* Increase the load beyond the expected user base to determine the system's breaking point.
* Observe how the system behaves under extreme conditions.
* Identify any unexpected failures or bottlenecks.

6. Scalability Assessment:

* Evaluate the system's ability to handle increased loads without proportional degradation in performance.
* Assess how the application scales with additional resources.

#### Tools Used:

* JMeter: For creating and executing performance tests.
* Monitoring Tools: (e.g., Grafana) for real-time monitoring of server resources during tests.
* Version Control System: (e.g., Git) to manage test scripts and configurations.