**Performance Test Plan for the PhotoAlbum web application**

**1. Objectives**

* Assess the website's ability to handle a normal load (Load Testing).
* Determine the website's behaviour under extreme conditions (Stress Testing).
* Evaluate the speed of the website's response to user actions (Response Time Testing).
* Measure and document performance metrics: Response Time, Error Rate, Latency.

**2. Scope**

**Load Testing**

* **Goal**: Test the maximum number of users the website can handle under normal conditions without degradation in performance.
* **Approach**:
  + Simulate an increasing number of concurrent users (e.g., 50, 100, 500, 1000 users).
  + Monitor how well the website handles the load and identify the threshold at which performance starts degrading.
* **Metrics to Measure**:
  + **Response time**: Time taken for pages (Home, User, Album, Photo) to load as more users access the system.
  + **Error rate**: Monitor if any requests fail as user load increases.
  + **System resource usage**: CPU, memory, and bandwidth utilization at peak loads.

**Stress Testing**

* **Goal**: Test how the website behaves when subjected to extreme conditions beyond normal operating limits (e.g., very high user load or resource exhaustion).
* **Approach**:
  + Simulate a sudden spike in users (e.g., from 1000 to 5000 users).
  + Test system limits by increasing traffic until the system fails, noting where it breaks down.
  + Check how the system recovers after the load returns to normal.
* **Metrics to Measure**:
  + **Response time**: How much longer it takes to load pages under extreme conditions.
  + **Error rate**: Track how many requests fail under overload.
  + **System behaviour**: Observe if the system crashes, slows down, or rejects requests.

**Response Time Testing**

* **Goal**: Measure how fast the website responds to user requests under normal and stressed conditions.
* **Approach**:
  + Measure the time it takes for key actions like login, viewing albums, or fetching photos.
  + Test response time with different numbers of users (e.g., 50, 200, 1000 users).
  + Identify actions that take too long and analyse possible bottlenecks.
* **Metrics to Measure**:
  + **Response time**: Time taken from the user request to the system's response.
  + **Latency**: Time delay before the data transfer starts.

**3. Test Environment**

* **Hardware**: Servers hosting the website, database server.
* **Software**: Load testing tools (e.g., JMeter, LoadRunner), monitoring tools.
* **Network**: Simulate different network conditions (e.g., good, average, poor bandwidth).

**4. Tools**

* **Apache JMeter**: To simulate concurrent users and monitor performance.
* **Postman**: To measure response times for API requests.
* **New Relic or Datadog**: To monitor system resource utilization (CPU, memory, network bandwidth) and latency.

**5. Performance Metrics**

* **Response Time**: Time taken to respond to user requests (measured in milliseconds/seconds).
* **Error Rate**: Percentage of requests that result in an error
* **Latency**: Delay before a data transfer begins (measured in milliseconds).
* **Throughput**: Number of requests processed per second.
* **System Resource Usage**: CPU, memory, and bandwidth usage under different loads.

**6. Acceptance Criteria**

* **Load Testing**: System should handle the defined maximum load without response times exceeding 2 seconds and an error rate below 1%.
* **Stress Testing**: System should degrade gracefully under extreme load without crashing, with a maximum error rate of 10%.
* **Response Time Testing**: All key user actions should complete within an acceptable time (e.g., < 500ms for login, < 1 second for fetching data).