**Performance Test Plan**

This is how I would structure the test plan

**1. Objectives**

The objective of this performance test plan is to evaluate the web application's performance by measuring its response time, error rate, and latency under varying user loads. This includes determining how the application behaves under both normal and extreme conditions.

**2. Scope of Testing**

1. **Load Testing**:
   * **Goal**: Determine the maximum number of concurrent users the application can handle under normal conditions without significant performance degradation.
   * **Approach**: Gradually increase the user load, starting from a baseline number of users and incrementing until the system’s response time or error rate becomes unacceptable.
2. **Stress Testing**:
   * **Goal**: Evaluate how the application behaves under extreme conditions, such as peak user load or when resources are limited.
   * **Approach**: Exceed the expected maximum user load to see how the system handles situations where resources are strained, focusing on observing system failures and recoveries.
3. **Response Time Testing**:
   * **Goal**: Measure the time it takes for the application to respond to various user requests under typical load conditions.
   * **Approach**: Conduct a series of requests to key pages (e.g., Home, User, Album, Photo) and measure the time taken for the server to respond and render each page.

**3. Performance Metrics**

The following metrics will be used to evaluate the application's performance:

* **Response Time**:
  + The time it takes for the application to respond to a request from the client side.
  + Acceptable threshold: <2 seconds for page load under normal conditions.
* **Error Rate**:
  + The percentage of failed requests during the test.
  + Acceptable threshold: <1% error rate under normal load conditions.
* **Latency**:
  + The delay before data transfer begins following a request.
  + Acceptable threshold: <100 ms under normal conditions.

**4. Execution Plan**

1. **Tools**:
   * For load and stress testing, tools like **Apache JMeter** or **Locust** can be used.
   * For response time and latency measurements, **Google Lighthouse**, **JMeter**, or browser developer tools can be utilized.
2. **Test Scenarios**:
   * **Load Testing Scenario**:
     + Begin with a baseline of 50 concurrent users and increase in increments of 50 every minute.
     + Monitor response time, latency, and error rate at each level until performance metrics degrade or reach an unacceptable level (e.g., response time exceeds 2 seconds or error rate exceeds 1%).
   * **Stress Testing Scenario**:
     + Start at the known maximum load from load testing results and increase the number of users by some figure at each stage.
     + Continue increasing the load until the application fails or crashes. Monitor for recovery time and check if the application can resume normal operations.
   * **Response Time Testing Scenario**:
     + Conduct tests for individual user actions, such as logging in, navigating between pages, and searching within albums.
     + Measure response time for each action and analyze latency.
     + Ensure response times are consistently below the threshold (2 seconds).

**5. Reporting and Analysis**

1. **Reporting Metrics**:
   * **Graphical representation** of response times, latency, and error rate against user load.
   * **Identify bottlenecks**: Indicate points at which performance degraded significantly.
   * **Summary**: Document findings, including which thresholds were exceeded and under what conditions, along with suggestions for improvement.
2. **Performance Analysis**:
   * Compare results against the acceptable thresholds.
   * Analyze results to determine if specific components or pages are causing delays.
   * Provide recommendations based on findings (e.g., increase server capacity, optimize database queries, or improve caching strategies).

Test cases based on my proposed test plan:

**Load Testing**

| **Test Case ID** | **Test Case Description** | **Steps** | **Expected Result** | **Metrics Captured** |
| --- | --- | --- | --- | --- |
| LT-01 | Verify application performance with 50 concurrent users | 1. Simulate 50 concurrent users accessing the Home page. | Response time is within acceptable limits (≤ 2 seconds). | Response Time, Error Rate, Latency |
| LT-02 | Verify application performance with 100 concurrent users | 1. Simulate 100 concurrent users accessing the Home and User pages. | Response time remains ≤ 2 seconds. Error rate is below 1%. | Response Time, Error Rate, Latency |
| LT-03 | Verify application performance with 200 concurrent users | 1. Simulate 200 concurrent users accessing all pages (Home, User, Album, Photo). | Application should handle the load without crashing. Response times may increase, but no more than 2 seconds. | Response Time, Error Rate, Latency |
| LT-04 | Determine maximum stable load | 1. Incrementally increase concurrent users by 50 until response times exceed acceptable limits or error rates increase. | Identify the maximum stable load before performance degradation. | Response Time, Error Rate, Latency |

**Stress Testing**

| **Test Case ID** | **Test Case Description** | **Steps** | **Expected Result** | **Metrics Captured** |
| --- | --- | --- | --- | --- |
| ST-01 | Evaluate system behavior at 120% of maximum stable load | 1. Increase user load to 120% of the previously determined maximum stable load. | Application may degrade, but should not crash. Response time may exceed 2 seconds, with a slight increase in error rate. | Response Time, Error Rate, Latency |
| ST-02 | Evaluate system behavior at 150% of maximum stable load | 1. Increase user load to 150% of the maximum stable load. | Application likely to experience significant performance issues; monitor for potential crashes. | Response Time, Error Rate, Latency |
| ST-03 | Evaluate system behavior at 200% of maximum stable load | 1. Increase user load to 200% of the maximum stable load. | Application should exhibit errors, possible crashes. Monitor for recovery after load decreases. | Response Time, Error Rate, Latency |
| ST-04 | Test recovery after extreme load conditions | 1. After reaching a breaking point (150% or 200%), gradually reduce load to normal levels and observe system recovery. | System should return to normal operational status with acceptable response times and minimal errors. | Response Time, Error Rate, Latency |

**Response Time Testing**

| **Test Case ID** | **Test Case Description** | **Steps** | **Expected Result** | **Metrics Captured** |
| --- | --- | --- | --- | --- |
| RT-01 | Measure response time for Home page navigation | 1. Access the Home page. 2. Capture time taken to load the page completely. | Page should load within 2 seconds. | Response Time, Latency |
| RT-02 | Measure response time for User page navigation | 1. Access the User page. 2. Capture time taken to load the page and display user data. | User data should load within 2 seconds. | Response Time, Latency |
| RT-03 | Measure response time for Album page navigation | 1. Access the Album page. 2. Capture time taken to load the page and display album data. | Album data should load within 2 seconds. | Response Time, Latency |
| RT-04 | Measure response time for Photo page navigation | 1. Access the Photo page. 2. Capture time taken to load the page and display photo thumbnails. | Photo thumbnails should load within 2 seconds. | Response Time, Latency |
| RT-05 | Validate response time for search functionality on Album page | 1. Enter a search term in the search box on the Album page. 2. Capture the time taken for search results to display. | Search results should display within 2 seconds. | Response Time, Latency |