Performance Engineer Home Task

Please find my answers highlighted in blue.

1. Explain the solution

- The test solution is created using Jmeter. This was agreed with Kalidass during the interview. I have not worked on K6. If required I can try to create a test script in K6. I would need some additional time to complete using K6.
- I have created a test script to measure performance metrics of the demo site - (https://blazedemo.com)
- Test Script Design and Steps
    - Landing Page
    - Enter departure and destination city and click on *"Find Flights"*
        - *Random cities will be picked from a data file*
    - Choose a flight.
    - Enter personal details and click on *"Purchase Flight".*

    - Assertions on each step to validate if the page doesn't have any errors.
    - User defined variables for following test inputs:
        - Environment – e.g., Prod, test, QA
        - Total Users
        - Ramp up duration
        - Test duration
    - Below is the default setting of the test script
        - Environment=Prod
        - Total Users=1000
        - Ramp up duration=15
        - Test duration=600 secs
    - To start test
        - Install Jmeter.
        - To open the script in edit mode run the following command in terminal ***<pathtojmeter>bin/jmeter -t AlphaSense-Assignment.jmx***
        - To start a test using default setting run following command in terminal ***<pathtojmeter>bin/jmeter -n -t AlphaSense-Assignment.jmx -l Run1.jtl -e -o Run1-1000U-Prod***
        - To view report open index.html from folder ***Run1-1000U-Prod***

- As mentioned in the home task, the first thread group named "load test" is designed to simulate 1000 users ramping up in 15 secs. The test will run for a duration of 10 mins.
- Apart from a load test, I have included a test plan for Soak Test, Stress Test and a Spike Test.

2. Analyse few HTTP/S responses in details

- Kindly note, I ran a test for 2 mins due to restrictions on total request/sec on the demo site. With high request/sec the site was giving restriction errors.

- Following are my observations of the test ran for a duration of **2 mins and 50 Users**.
- Almost all the responses received were HTTP 200 - Success. There was only 1 response which returned HTTP 500 – Internal Server Error
- All the assertions passed (except for 1 mentioned above). Error rate 0.01%.
- The average throughput achieved for each page was ~27.4/sec with a combined total throughput of ~108/sec
- The response time for each page was ~550-705 ms. There were few spikes of max response time = 3.5 secs
- The hits/sec and transaction/sec graphs are linear. This means increase in both the hits/sec and transaction/sec were in direct proportion to increase in load.
- Conclusion – Based on above observations, the test was a PASS. I would run few more rounds with varying load profiles and longer duration to get more insights on performance metrics.

3. Did the load test have an impact on web application response time?
    - Kindly note, as this is a demo site, there were restrictions from the site on the request/sec rate.
    - By observing the test for duration of 2 mins and load of 50 users, there were no impact on response time by the load test.
    - However, when the throughput rate was increased, there were lot of HTTP 500 errors
4. What is the optimal application response time for modern web applications?
    - For an API - Response time should be within 200-300 ms.
    - For a full browser render of a page – Response time should be within 3-4 sec. Hero elements to be displayed within 1 sec.
    - General human perception - people start to lose interest in the >= 6 second range, by 10-15 seconds you have typically lost them, unless you REALLY have something they want or need
5. How would you define acceptable load for web applications?
    - I usually follow the 80/20 rule while finalising the test scenarios. This means, 80% of the load is generated by 20% of the business flow.
    - I would draw a baseline number for average load over past 3 months of high user activity days. I will also get an estimate of the future projection of the user load by discussing with Business Analyst or PO.
    - Based on the above calculations, I will create test profile for 1x, 2x, 2.5x average load. The acceptable load to make the application future proof and scalable will be 2.5 times the average load.