1. Explain the process of model quantisation and discuss its benefits and potential drawbacks in AI model optimisation. (5 Marks)

Model quantisation is basically about making an AI model lighter by reducing the precision of its numbers. For example, instead of storing weights as 32-bit floats, we can store them as 8-bit integers. This makes the model smaller and lets it run faster.

The main benefits are lower memory use, faster inference, and less power consumption. For example, a quantised model can be run on a mobile app for faster speed and lower battery consumption.

The drawback is that reducing precision can hurt accuracy, and some models are more sensitive than others. Sometimes some extra fine-tuning is needed to keep performance stable.

2. Discuss the role of caching in scaling applications. Compare and contrast in-memory caching and distributed caching, providing examples of when each would be most appropriate. (5 Marks)

Caching helps apps scale by keeping frequently used data close at hand so the system doesn't always need to hit the database. This speeds things up and reduces load.

There are two main approaches:

- In-memory caching (e.g. Redis) keeps data in RAM on a single server. It's very fast and works well for user sessions. The limitation is it only scales as far as that one machine's memory.
- Distributed caching spreads cached data across multiple servers. This is a bit slower than in-memory, but it scales much better and stays reliable if one server goes down. It's useful for large, high-traffic systems like global e-commerce platforms.

3. Explain the concept of horizontal scaling and vertical scaling. Provide examples of when you would choose one over the other in a real-world scenario. (5 Marks)

- Vertical scaling means upgrading a single machine with more CPU, RAM, or storage. For example, giving a database server extra memory so it can handle bigger queries.
- Horizontal scaling means adding more machines or servers to share the load, like putting a load balancer in front of multiple web servers.

- If an e-commerce site is getting slightly heavier traffic, it might be cheaper and simpler to just vertically scale by adding more RAM to the database. But if that site suddenly grows globally with millions of users, horizontal scaling makes more sense. This means addingg more servers behind a load balancer to handle the distributed traffic.

4. Describe the process of designing an effective load test for a web application. What key factors should be considered, and how would you determine the success criteria for the test? (5 Marks)

Designing an effective load test begins by defining the goals, such as the number of concurrent users the system should support. The process involves modelling realistic workloads, creating test scenarios that reflect common user actions, and setting up a staging environment similar to production. During the test, key metrics like response time, throughput, error rate, and resource usage should be monitored. Tools such as JMeter, Locust, or k6 can be used. Success is determined by clear criteria, for example, ensuring that 95% of requests complete within two seconds and error rates remain below 1%.