INTERVIEW QUESTIONS – GENERATIVE AI FOR EMBEDDED SYSTEMS

SECTION 1: AI / ML / GENERATIVE AI FUNDAMENTALS

1. What is the difference between Artificial Intelligence, Machine Learning, Deep Learning, and Generative AI?
2. Why is Generative AI different from traditional machine learning models?
3. What are discriminative models and generative models? Give examples.
4. Why are autoencoders considered generative models?
5. What types of embedded or IoT problems are suitable for Generative AI?
6. Why is accuracy not always a meaningful metric in anomaly detection?
7. Why are labels not required in autoencoder-based anomaly detection?
8. What limitations do embedded systems impose on Generative AI models?

SECTION 2: MACHINE LEARNING METHODS

9. Compare classification, regression, and clustering with embedded system examples.
10. When would you choose classification instead of anomaly detection?
11. Why is clustering considered an unsupervised learning technique?
12. Explain how regression can be used in IoT or sensor-based applications.
13. Why is classification unsuitable for detecting unknown faults?
14. Can clustering be used for anomaly detection? If yes, how?

SECTION 3: LIGHTWEIGHT GENERATIVE MODELS (AUTOENCODERS)

15. Explain the architecture of an autoencoder.
16. Why must the input and output dimensions of an autoencoder be the same?
17. What is reconstruction error?
18. Why is Mean Squared Error commonly used in autoencoders?
19. How does an autoencoder behave when it receives unseen abnormal data?
20. What happens if an autoencoder is trained on both normal and abnormal data?
21. Why are small dense neural networks preferred on microcontrollers?
22. How do you decide the latent space size in an autoencoder?

SECTION 4: TIME-SERIES AND EMBEDDED DATA HANDLING

23. What is time-series data and why is it common in embedded systems?
24. When is it acceptable to remove timestamps from sensor data?
25. Is a sliding window always required for time-series anomaly detection?

26. Compare point-wise anomaly detection and window-based detection.
27. Why is fixed-range normalization preferred in embedded ML systems?
28. Which preprocessing steps must remain identical during training and inference?

SECTION 5: EMBEDDED AI DEPLOYMENT (STM32 / EDGE AI)

29. What is TensorFlow Lite and why is it used for embedded systems?
30. What is the role of X-CUBE-AI in STM32 development?
31. Explain the input and output tensor shapes of an autoencoder deployed on STM32.
32. What is the activation buffer and how is its size determined?
33. What does n_batches mean in STM32 AI inference?
34. Why are preprocessing and postprocessing done outside the neural network on MCUs?
35. How do you verify that the correct AI model is running on embedded hardware?
36. Why is INT8 quantization optional but not mandatory?

SECTION 6: ETHICS, SAFETY, AND INDUSTRY PRACTICES

37. What ethical challenges arise when using AI in embedded or industrial systems?
38. Why is explainability important in embedded AI applications?
39. How does energy efficiency influence AI model design choices?
40. Why is deterministic behavior preferred in embedded AI systems?
41. What risks arise from deploying overfitted AI models in safety-critical systems?
42. How would you validate an embedded AI system before production deployment?

SECTION 7: CAPSTONE AND PROJECT-ORIENTED QUESTIONS

43. Explain the end-to-end workflow of a Generative AI embedded project.
44. What challenges do you face when converting a model to TensorFlow Lite?
45. How do you test an embedded AI model with new unseen sensor data?
46. How would you integrate an embedded AI solution with IoT or cloud services?
47. If memory or CPU budget is reduced, what changes would you make to your model?