

Q1. What is Qt and what primary problem does it solve in the Desktop space?

Answer:

Q2. Explain the core architecture or components of Qt and how they work together.

Answer:

Q3. What are the main alternatives to Qt and when would you choose it over them?

Answer:

Q4. How do you install, configure, and get started with Qt for a new project?

Answer:

Q5. What are the key configuration options or settings that affect Qt's behavior in production?

Answer:

Q6. How does Qt handle scalability, reliability, and high availability?

Answer:

Q7. What observability does Qt provide: metrics, logs, traces, or health endpoints?

Answer:

Q8. What are the common performance bottlenecks in Qt and how do you address them?

Answer:

Q9. What security best practices should you follow when deploying Qt?

Answer:

Q10. What are the most common pitfalls or mistakes teams make when adopting Qt?

Answer:
