# Technical Feedback questions - EXAMPLE

**🔧 Code Performance**

1. Are there any performance bottlenecks or inefficiencies in the current codebase?
2. Have you noticed any areas where the code could be optimized (e.g., memory usage, loop efficiency, redundant operations)?
3. Are we following best practices for asynchronous operations and concurrency?
4. How does the current implementation handle large datasets or high-throughput scenarios?
5. Do you see opportunities to refactor the code for better runtime performance?

**🔐 Security**

1. Are there any obvious or potential security vulnerabilities in the code (e.g., input validation, SQL injection, XSS)?
2. Are we properly handling authentication, authorization, and session management?
3. Is sensitive data (e.g., passwords, tokens, user info) being securely stored and transmitted?
4. Does the prototype rely on any third-party libraries that could introduce security risks?
5. Are we following secure coding standards (e.g., OWASP Top 10)?

**📈 Scalability**

1. Can the current architecture handle an increase in users or traffic without major rework?
2. Are components modular and decoupled enough to scale independently (e.g., microservices, load balancing)?
3. Are there any parts of the code or infrastructure that could become bottlenecks at scale?
4. How does the system manage state and caching in a distributed environment?
5. Are we prepared for horizontal scaling (e.g., database sharding, autoscaling in cloud environments)?

**🛠️ Use of Tools – GitHub / Code Review**

1. Is the codebase well-organized and easy to navigate on GitHub?
2. Are pull requests (PRs) reviewed thoroughly with meaningful comments and feedback?
3. Are we using GitHub features effectively (e.g., Issues, Projects, Actions, Code Owners)?
4. Is version control used consistently and according to branching strategies (e.g., Git Flow, trunk-based development)?
5. Do we have automated checks and CI/CD pipelines integrated with GitHub for quality control?