**What tests should be implemented in the programming of such a project?**

Unit tests for core logic, integration tests for API endpoints, and end-to-end tests for the UI should be implemented. Additionally, performance tests should be conducted to ensure the reliability of OSINT tool execution and data handling.

**What tests would you include in your project if given more time?**

I would add unit, integration, and performance tests to evaluate system behaviour under heavy load. Also, I have already conducted end-to-end tests by verifying how the UI interacts with the backend.

**How can you measure the performance of your project? Can you optimize your code to run faster?**

Performance can be measured using logging, metrics collection (Prometheus, Grafana), and profiling tools to monitor response times and resource usage. I have already implemented asynchronous processing to handle long-running scans efficiently, and further optimization can be done by improving query performance in PostgreSQL, adding indexes at the database level for faster lookups, and implementing caching to reduce redundant data fetching.

**What bottlenecks can be caused by the OSINT tools given to this project?**

OSINT tools like Amass and theHarvester can cause high CPU and memory usage, leading to slow processing. Additionally, network latency, API rate limits, and reliance on external data sources can delay data retrieval, affecting scan completion times and resulting in varying scan outputs.