

SRE 1

1. How to measure Accessibility:

Approach:

- Use Response Time as a key indicator for accessibility.
- Employ tools that simulate user interactions and measure the time taken for the application to respond.
- Calculate the percentage of successful interactions within a defined time frame.

Justification:

- Faster response times correlate with better user experience and perceived accessibility.
- An SLI for Accessibility could be set at 95% of interactions completing within 2 seconds.

2. How to measure Mean Downtime:

Approach:

- **Downtime Monitoring**
 - **Method:** Use monitoring tools to track and log instances of downtime.
 - **Justification:** Monitoring tools provide real-time insights into system performance and downtime events. Logging downtime instances helps calculate the mean downtime over a specific period.
- **Incident Reporting:**
 - **Method:** Encourage a robust incident reporting system where team members document and timestamp downtime incidents.
 - **Justification:** Incident reports provide detailed information on the duration and impact of each downtime event. Aggregating this data over time allows for the calculation of mean downtime.

3. Factors to decrease Mean Downtime:

Actions:

- Implement automated failover mechanisms to minimise manual intervention during incidents.
- Conduct regular chaos engineering exercises to proactively identify and address weaknesses in the system.
- Optimise deployment processes to reduce the impact of releases on production.

Justification:

- Automation reduces the time required for manual interventions, speeding up incident resolution.
- Chaos engineering helps identify and fix potential issues before they impact users.
- Optimised deployment processes minimise the likelihood of introducing new problems during releases.

4. Identify Service Level Objectives and Indicators:

SLOs:

- Achieve 99.9% availability for the application.
- Maintain an average response time under 200 milliseconds

Justification:

- SLOs provide a clear target for service performance aligned with user expectations.
- SLIs help monitor and measure the performance of critical components, ensuring they meet SLOs.

5. Risk Acceptance and Mitigation Plan

Risk Description	Likelihood	Impact	Severity	Mitigation Strategy
Certain team members have limited experience with chosen tech stack	Medium	Medium	Medium	Conduct a brief technical briefing session at the start of the hackathon to ensure everyone is on the same page. Encourage collaboration and knowledge-sharing within the team.
Lack of effective communication within the team	Medium	Medium	Medium	Set up regular check-ins throughout the hackathon. Use collaboration tools such as messaging platforms and video calls. Foster an open and supportive team environment.
Inadequate time management during the hackathon	High	High	High	Establish a clear timeline with milestones. Prioritize tasks based on complexity and importance. Periodically review progress and adjust the plan if necessary.
Lack of version control leading to code conflicts	Medium	Medium	Medium	Implement version control practices (e.g., Git) and ensure all team members are familiar with the basics. Regularly commit and pull changes to avoid conflicts.
Insufficient testing resulting in a buggy application	Medium	High	High	Allocate dedicated time for testing. Implement test-driven development (TDD) practices. Use automated testing tools where possible.
Insufficient Error logging and handling	Medium	High	Medium	Implement robust error handling mechanisms. Ensure proper logging is in place to facilitate debugging and troubleshooting.
Integration Challenges between Frontend and Backend	Medium	High	High	Establish regular communication channels between frontend and backend teams. Conduct joint planning sessions to align on data formats, API contracts, and overall system architecture.