**Section 1: Healinium Basics (Q1–Q20)**

1. **What is Healinium?**  
   An AI-powered locator healing tool for Selenium/WebDriver that automatically fixes broken locators.
2. **Why use Healinium?**  
   To prevent test failures when the UI changes.
3. **Main goal of Healinium?**  
   Increase automation test script stability.
4. **Which frameworks can Healinium be used with?**  
   Selenium, Appium, and WebDriver-based frameworks.
5. **How does Healinium fix locators?**  
   Uses AI algorithms and element attribute comparisons.
6. **Which languages does Healinium support?**  
   Java, C#, Python (any language supported by Selenium).
7. **Relation between Healinium and Selenium?**  
   Healinium acts as an extension to Selenium.
8. **Does Healinium run standalone?**  
   No, it integrates into a Selenium framework.
9. **What types of locators can Healinium handle?**  
   XPath, CSS, ID, Name, Class Name, Link Text, etc.
10. **What is locator healing?**  
    Process of automatically applying a new locator when the old one fails.
11. **Architecture of Healinium?**  
    Test Script → Healium Wrapper → AI Healing Engine → Selenium Driver.
12. **Why is Healinium considered intelligent?**  
    It analyzes element attributes to find the best possible match.
13. **Learning process of Healinium?**  
    Stores and learns from previous locator and attribute history.
14. **Can Healinium handle dynamic elements?**  
    Yes, even when attributes change.
15. **Accuracy of Healinium results?**  
    Typically 90%+ when attributes are available.
16. **Which browsers does Healinium work with?**  
    All Selenium-supported browsers (Chrome, Firefox, Edge, etc.).
17. **Does Healinium work for mobile automation?**  
    Yes, via Appium integration.
18. **Is manual locator update needed with Healinium?**  
    Usually no, but may be required in critical cases.
19. **Does Healinium reduce test execution speed?**  
    Slightly, due to the healing process.
20. **What is Healinium’s license type?**  
    Depends on version — could be free or paid.

**Section 2: Healinium Setup & Integration (Q21–Q40)**

1. **How to install Healinium?**  
   Add Maven/Gradle dependency.
2. **How to add Healinium dependency in Maven?**  
   Add Healinium coordinates in pom.xml.
3. **How to add Healinium in Gradle?**  
   Add dependency in build.gradle.
4. **What is the Healinium wrapper driver?**  
   HealiumDriver — a replacement for Selenium WebDriver with healing support.
5. **How to initialize HealiumDriver?**  
   WebDriver driver = new HealiumDriver(new ChromeDriver());
6. **What is Healinium config file?**  
   File that defines healing behavior and storage.
7. **Healing storage options?**  
   Local file, database, cloud.
8. **What is Healinium Local Mode?**  
   Stores healing data locally.
9. **What is Healinium Remote Mode?**  
   Stores healing data on a central server/cloud.
10. **Where are Healinium logs displayed?**  
    Console and log files.
11. **Does Healinium support XML configuration?**  
    Yes, supports properties or XML.
12. **Default storage format for Healinium?**  
    JSON or properties-based.
13. **Does Healinium driver support implicit waits?**  
    Yes.
14. **Does Healinium work with explicit waits?**  
    Yes, integrates with wait conditions.
15. **Does Healinium work with Page Object Model (POM)?**  
    Yes, locators in POM are healed too.
16. **Does Healinium handle Shadow DOM?**  
    Limited support, similar to Selenium.
17. **Can Healinium be integrated with CI tools?**  
    Yes, works with Jenkins and other CI pipelines.
18. **How to check Healinium version compatibility?**  
    Check official docs or Maven Central.
19. **What is a backup locator in Healinium?**  
    Locator used if the primary one fails.
20. **How to set Healinium timeout?**  
    Through config file or driver initialization.

**Section 3: Healinium Features & Benefits (Q41–Q60)**

1. **Main benefit of auto-healing locators?**  
   Reduces test instability and maintenance costs.
2. **Can Healinium heal both CSS and XPath?**  
   Yes.
3. **Does Healinium support multiple locator strategies?**  
   Yes, with primary and fallback.
4. **How does Healinium match attributes?**  
   Calculates similarity score.
5. **What is similarity score?**  
   Percentage match between old and new element attributes.
6. **What is minimum similarity threshold?**  
   Minimum percentage to accept a match.
7. **Can threshold be changed in Healinium?**  
   Yes, via config.
8. **Does Healinium keep locator history?**  
   Yes, for accurate future healing.
9. **Does Healinium generate reports?**  
   Yes, with healed locators summary.
10. **What does Healinium report show?**  
    Broken locator, healed locator, match score.
11. **Does Healinium prevent test failures?**  
    In most cases, yes.
12. **Can Healinium handle page load delays?**  
    Yes, using waits.
13. **Does Healinium support multi-language scripts?**  
    Only languages supported by Selenium.
14. **Can Healinium handle element index changes?**  
    Yes, by using other attributes.
15. **Difference between Healinium and smart XPath?**  
    Healinium heals at runtime, smart XPath optimizes at design time.
16. **Does Healinium have self-learning?**  
    Yes, improves accuracy with repeated healing.
17. **Can you manually override Healinium healing?**  
    Yes, in config or code.
18. **Can Healinium handle dynamic IDs?**  
    Yes, by comparing other attributes.
19. **Limitations of Healinium?**  
    May fail if element is completely new.
20. **Effect of Healinium on regression tests?**  
    Reduces false failures.

**Section 4: Healinium Best Practices & Troubleshooting (Q61–Q80)**

1. **What to check first when starting Healinium?**  
   Storage location and threshold.
2. **Optimal Healinium threshold?**  
   0.7–0.8 (70–80%).
3. **How to integrate Healinium with CI/CD?**  
   Add dependency in Maven build and Jenkins pipeline.
4. **Why store locators in POM when using Healinium?**  
   For better maintainability.
5. **Why is dynamic wait important in Healinium?**  
   To handle slow-loading pages.
6. **When to use cloud storage in Healinium?**  
   For sharing healing data across teams.
7. **How does Healinium store in database?**  
   Using JDBC connection config.
8. **What is Healinium debug mode?**  
   Shows detailed healing decision logs.
9. **What to do if test fails in Healinium?**  
   Check logs and update locator manually if needed.
10. **When can performance issues occur in Healinium?**  
    With large DOM and high healing threshold.
11. **What is ElementMatchException?**  
    No matching element found.
12. **How to fix locator mismatch problems?**  
    Clear locator history.
13. **When is backup locator used?**  
    If primary locator’s score is too low.
14. **Does Healinium help with dynamic tables?**  
    Yes, even if column/index changes.
15. **Can healing be disabled in Healinium?**  
    Yes, in config.
16. **Can attribute weights be adjusted in Healinium?**  
    Yes, by changing attribute priority in config.
17. **What is element cache in Healinium?**  
    Stores recently healed elements.
18. **How to enable continuous learning in Healinium?**  
    Use central storage.
19. **What is manual approval process in Healinium?**  
    Approve healing results before saving.
20. **How to show Healinium execution reports in CI?**  
    Publish report path in Jenkins.

**Section 5: Healinium Advanced Usage (Q81–Q100)**

1. **Does Healinium support parallel execution?**  
   Yes, with thread-safe config.
2. **Does Healinium sync locators across browsers in multi-browser tests?**  
   Yes, if central storage is used.
3. **Can Healinium heal mobile app locators?**  
   Yes, via Appium.
4. **Does Healinium support language switching?**  
   Locators are language-independent.
5. **Can Healinium be used in API testing?**  
   No, UI automation only.
6. **Does Healinium have full Shadow DOM support?**  
   Partial only.
7. **Can Healinium heal iframe elements?**  
   Yes, after switching frames.
8. **Can page load strategy be set in Healinium?**  
   Yes, via Selenium settings.
9. **Can Healinium handle non-HTML apps?**  
   No, HTML-based UI only.
10. **Can Healinium heal SVG elements?**  
    Yes, by matching attributes.
11. **What to check during version upgrade in Healinium?**  
    Config compatibility.
12. **Can custom matchers be added in Healinium?**  
    Yes, by extending API.
13. **Does Healinium support headless mode?**  
    Yes, same as Selenium.
14. **Does Healinium support remote execution?**  
    Yes, compatible with Selenium Grid.
15. **What is cloud locator service in Healinium?**  
    Centralized locator storage service.
16. **Does Healinium allow user-defined locator strategy?**  
    Yes.
17. **Does Healinium affect test data?**  
    No, works only at locator layer.
18. **Can healing results be exported from Healinium?**  
    Yes, to JSON/CSV.
19. **How to restore backups in Healinium?**  
    Restore storage file or DB.
20. **Overall goal of Healinium?**  
    Reduce locator maintenance and improve automation reliability.