1. What is Self-Healing Test Automation?

A. Self-Healing in test automation means identifying changes made to web elements on a web page, such as text fields, buttons, dropdowns etc. Using self-healing we can automatically detect these changes in web elements and adapt the test execution accordingly, without any errors or human intervention during the test execution.

2. What is Healenium?

* Healenium is an open-source library for automated testing that leverages the power of ML to improve the reliability of tests.
* It automatically detects and heals test failures caused by changes in the UI, such as element IDs or class names.
* This reduces the time and effort required to maintain automated tests.

3. Healenium Components:

* Healenium-Web: Maven Dependency.
* Healenium-Backend: Dockerized Container Service through images.

4. Steps involves in configuring Healenium-Web, Healenium-backend, and Healenium-db with Project.

Pre-reqs:

* Make sure you have the recent Docker version installed.
* Install any Java IDE.

Steps:

* Create a new Java-Selenium Maven project.
* Add Healenium-Web and Selenium dependencies in pom.xml file.
* Create healenium.properties file under resources folder.
* Setup Healenium-db: Create an infra folder under the project, a db folder inside it, and an sql folder inside it where we store the init.sql file created through a cmd line.
* Setup Healenium-Web with Docker: Go to the infra folder and using a curl cmd, we get the docker-compose.yaml file. Then run that yaml file using ‘docker-compose up -d’
* We can verify if the ‘Healenium-backend’ container and the ‘Healenium-db’ container both are running, on Docker Desktop.

5. What is the healenium.properties?

A. We configure different parameters for helenium:

1. recovery-tries: How many times the elements will be tried to be healed.
2. score-cap: The minimum matching score required for the detected locator to be accepted.
3. heal-enable: To enable/disable element healing.
4. serverHost: Host on which healenium-backend container runs.
5. serverPort: Port on which healenium-backend container runs(7878 by defualt).
6. imitatePort: Port on which selector-imitator instance runs.

6. How to use a Healenium driver in the Selenium Testcase?

A. Create a normal WebDriver object, then use ‘SelfHealingDriver.create(<driver\_object>) and pass the WebDriver object to create a SelfHealingDriver object.

Eg: WebDriver delegate = new ChromeDriver();

SelfHealingDriver driver = SelfHealingDriver.create(delegate);